EARLY ADMISSION SELECTION PROCESS INTO SIXTH FORM SCIENCE STREAMS USING NEURAL NETWORKS MODEL

A thesis submitted to the Graduate School in partial
fulfilment of the requirements for the degree
Masters of Science (Information Technology)
University Utara Malaysia

by
Wong Tuck Sung

Sekolah Siswazah
(Graduate School)
Universiti Utara Malaysia

PERAKUAN KERJA KERTAS PROJEK
(Certification of Project Paper)

Saya, yang bertandatangan, memperakukan bahawa
(I, the undersigned, certify that)

WONG TUCK SUNG

calon untuk Ijazah
(candidate for the degree of) Sarjana Sains (Teknologi Maklumat)

telah mengemukakan kertas projek yang bertajuk
(has presented his/her project paper of the following title)

EARLY ADMISSION SELECTION PROCESS INTO SIXTH FORM
SCIENCE STREAMS USING NEURAL NETWORKS MODEL

seperti yang tercatat di muka surat tajuk dan kulit kertas projek
(as it appears on the title page and front cover of project paper)

bahawa kertas projek tersebut boleh diterima dari segi bentuk serta kandungan, dan meliputi bidang ilmu dengan memuaskan.
(that the project paper acceptable in form and content, and that a satisfactory knowledge of the field is covered by the project paper).

Nama Penyelia
(Name of Supervisor): Puan Fadzilah Siraj

Tandatangan
(Signature) : Fadzilah

Tarikh
(Date) : 30/09/2000

GRADUATE SCHOOL
UNIVERSITI UTARA MALAYSIA

PERMISSION TO USE

In presenting this thesis in partial fulfilment of the requirements for a postgraduate degree from Universiti Utara Malaysia, I agree that the Universiti Library may make it freely available for inspection. I further agree that permission for copying of this thesis in any manner, in whole or in part, for scholarly purposes may be granted by my supervisor(s) or, in their absence, by the Dean of the Graduate School. It is understood that any copying or publication or use of this thesis or parts thereof for financial gain shall not be allowed without my written permission. It is also understood that due recognition shall be given to me and to Universiti Utara Malaysia for any scholarly use which may be made of any materials from my thesis.

Requests for permission to copy or to make other use of materials in this thesis, in whole or in part, should be addressed to:

Dean of Graduate School
Universiti Utara Malaysia
06010 UUM Sintok
Kedah Darul Aman

ABSTRACT (BAHASA MALAYSIA)

Tujuan saya menyiapkan kajian ini adalah untuk menyiasat dan menyediakan satu model berasaskan Teori Rangkaian Neural yang berkemampuan memilih pelajar yang berkelayakan untuk kemasukan awal ke Tingkatan Enam dalam aliran sains. Ianya bertujuan untuk membolehkan proses pemilihan dilakukan sebelum keputusan Sijl Pelajaran Malaysia diumumkan. Matlamat utama kemasukan awal diadakan adalah untuk membolehkan para pelajar memulakan pelajaran lebih awal dan dapat menghabiskan sukatan pelajaran tepat pada waktunya. Motifnya adalah untuk menjimat dan mengelakkan pembaziran masa daripada hanya membiarkan pelajar menunggu hingga keputusan peperiksaan diumumkan sebelum mereka memulakan persekolahan.

Satu penyelesaian berasaskan rangkaian neural dengan menggunakan "Multi Layer Perceptron" dan "Steepest Gradient Descent" telah digunakan untuk menghasilkan model yang lebih baik dan terperinci untuk memilih pelajar. Sejumlah 1488 contoh data daripada sepuluh buah sekolah menengah di negeri Perak telah digunakan berdasarkan peperiksaan dalaman yang telah diambil oleh bekas palajar Tingkatan 4 dan Tingkatan 5. Data ini telah dikumpulkan supaya dapat dilatih dan diuji dengan menggunakan perisian Neuron Connection Version 2. Ramalan ketepatan iaitu 92.18% telah dicapai dengan menggunakan model rangkaian neural ini.

Hasil daripada analisa data tersebut menunjukkan satu korrelasi yang kukuh terjalin antara pembolehubah input, yang terdiri daripada markah matapelajaran, agregat dan gred, dengan pembolehubah output iaitu tawaran ke Tingkatan Enam. Ia juga menunjukkan bahawa data tersebut hanya sedikit condong (skewed) dan ditaburkan secara normal. Model rangkaian neural yang terlatih ini juga telah didapati menunjukkan ketepatan yang baik apabila disesuaikan kepada data-data lain, sama ada kepada para pelajar sekolah lelaki atau sekolah perempuan, mahu pun kepada sekolah kawasan bandar atau pun sekolah luar bandar.

ABSTRACT

The purpose of this study is to investigate and offer a model, based on Neural Networks Theory, capable of selecting successful students for early admission into sixth form science streams. This model would be able to perform the intended selection process even before the results of the Sijil Pelajaran Malaysia (SPM) are announced. The main benefit of this early admission was to allow students to start their classes early and to complete their demanding syllabus on time. The noble motive was to save time and to prevent time wasting, which would be true if students had to wait until the examination results are announced before they could start their classes.

A neural networks solution, using Multi Layer Perceptron (MLP) and Steepest Gradient Descent algorithm, was studied to offer a better model to select students more meticulously. A total of 1488 data samples from ten secondary schools in the silver state of Perak, consisting of past-year Form 4 and Form 5 internal examination results, were collected in order to be trained and tested using the Neural Connection Version 2 software. A correct prediction of 92.18% accuracy was achieved using this Neural Networks model.

Analysis of the data showed a reasonably strong correlation between the input variables, which consisted of subjects' marks, aggregates and grades achieved, with the targeted output variable, which was the offer to continue with Sixth Form. It also showed that the data were only slightly skewed and were normally distributed. The trained Neural Networks model was found to produce a comparable accuracy when applied to other data from either only boys or girls schools, and from either urban or rural schools.

ACKNOWLEDGEMENTS

Special thanks to my wife, Anna, whose wide experience in research had been a constant help. Our continuous discussions throughout the study of this thesis had helped me to find a clear focus.

TABLE OF CONTENTS

List of tables

List of figures

CHAPTER ONE:
INTRODUCTION

1.0 Background

Education has always been highly valued and often regarded as the secured passport for a bright future and as a sure mean towards self-improvement. Science-based education has been the focus of the education ministry lately, as was evident in the ongoing push to encourage more students to take up science stream in their secondary years. This was in line with the goal to fulfill the nation need for more knowledge workers to make Vision 2020 a success.

Much has been stated about the need to start Sixth Form as early as possible to prevent time wasting after students had finished taking their SPM examination. The time lapse between the Sijil Pelajaran Malaysia (SPM) examination and the start of Sixth Form used to be about 6 months. This had been necessary to allow enough time for the quality marking and accurate grading of the SPM examination papers, as well as to correctly select successful candidates to be placed in secondary schools to continue their Sixth Form education.

For this year, the Education Ministry has implemented a new system to start Sixth Form classes as early as possible. Classes started in the month of February instead of the usual month of May. It was reported in the Sunday Star's Education pullout dated 20 February, 2000 that.

> About 26,000 students have been given the opportunity to begin Sixth Form in February, three months ahead of the usual enrolment date. The move is to allow more time for the syllabus to be completed and to keep students gainfully occupied. (p. 2)

The selection of these successful students for this early intake had been based solely on the students' own school SPM trial examination results. Only first graders were selected while the rest will have to wait for the SPM results in April. Each school submitted a list of names of successful students to the State Education Department. Those who were not offered the early intake because of poor results in the SPM trial examination, but scored well in the SPM were then admitted as the second batch intake in June. These second group of students had to study harder to catch up with the first group of students. However, those that were admitted early but performed poorly in the real SPM examination had to vacate their places. It was reported in the New Straits Times on 21 June, 2000 that:

> Some 1,000 Form Six students could not continue their studies, as their Sijil Pelajaran Malaysia examination results in March did not fulfil the minimum entry requirements. (p. 8)

In order to be enrolled into Sixth Form science stream, the minimum entry requirements a student needed to have was to secure a credit in Bahasa Malaysia and at least three other credits from pure science (Physics, Chemistry and Biology) and mathematics (Additional Mathematics and Mathematics) subjects. A student must also secure an aggregate of between three and eighteen, taken from all three pure science subjects or from one mathematics and two science subjects.

However, some confusion was caused when some students who met all the minimum requirements to qualify for Form Six were not offered places as their names were left out of the list given to schools after the announcement of the SPM examination results. In response to this, in a report taken from The Star on 27 June 2000, the Education Ministry said that:

> ... all students who have met all the minimum requirements to qualify for Form Six will be given places. .... These students can appeal directly to the respective state departments and should get a response within a few days to a week. (p. 10)

In view of the problem mentioned above, a better and more accurate selection process is needed. With the impending intrusion of computer technology into most areas of our lives, a new method based on artificial intelligent technology is much needed to come out with a more acceptable and more accurate selection. Neural Networking, with its ability to recognize and classify complex patterns and trends in data, is the suggested answer to this problem.

1.1 Problem Definition

The Statement of the Problem

The selection of successful students into Sixth Form, in the past, has always been based on the student's SPM examinations' results and the availability of places in the various secondary schools. In the recent past, students used to be selected for admission only after the announcement of the SPM results that usually came out in April or May every year. Starting this year, students were pre-selected based on their school based trial examinations. To qualify for admission into the science stream, the criteria for selection are:

1. Pass SPM with at least 3 credits, including for Bahasa Malaysia.
2. Achieve at least 3 credits for science and mathematics subjects with aggregate less or equals to 18,
3. Age between 17 and 20 years.

Students scoring lower than the minimum requirements were allowed to appeal to be placed into Sixth Form. They could only do this after the SPM result was announced. However, their eventual successful admission depended on whether there were any places left in the various schools, or where there was a lowering of the minimum requirements as stated above.

Selecting students for early intake purportedly had been to ensure that students did not waste time during the long break but to continue studying in order to be able to complete the demanding Sixth Form syllabus in time. Though the intended goal was perfectly good, the method for selection left much to be desired. Those who missed out in the early intake but were successful in their SPM had an uphill task to catch up with their studies. Those who were in the early intake but who were not successful in their SPM would have to move out. For both these groups of students, time was still wasted. This was ironic as this new early intake was meant to prevent time wasting.

In order to come out with a better and more accurate selection process, and at the same time aimed at preventing time wasting for wrongly selected students, the researcher suggested Neural Networking as an attempt to solve this problem. In order to stay in focus during this study to offer a more viable and practical solution, it was the intention of the researcher to concentrate only on science based students who will be offered places into Sixth Form Science classes for this project. This project is thus aptly titled "**Early Admission Selection Process Into Sixth Form Science Streams Using Neural Networks Model.**"

1.2 Purpose of the Study

The purpose of the proposed study is to offer a neural networks model solution in the selection of successful students for early admission into Sixth Form science stream.

1.3 Research Questions

Some pertinent questions need to be studied carefully first to identify the correct factors that will influence the success of this project. The questions that need to be answered in this project are:

1. Is there a positive co-relation between a student's past school based examination results with his Sixth Form offer?
2. What subjects taken by the student during his Form 4 and Form 5 will determine correctly, to a certain acceptable percentage, his Sixth Form offer? (Note: This project intends to focus only on science students.)
3. How significant will the student's Form 4 examination results influence his Sixth Form offer?
4. How significant will the student's Form 5 examination results influence his Sixth Form offer?
5. Do other factors, like urban or rural school and gender, besides past examination results contribute positively towards a more accurate prediction on his Sixth Form offer?

CHAPTER TWO

LITERATURE REVIEW

2.0 Overview of Chapter

The aim of this chapter is to introduce to the reader the main concept of the neural networks theory, starting from the basics and moving on to how learning is done in neural networks. A simple model of an artificial neuron was also discussed, leading to the introduction of the multi layer perceptron that was used in this project. The three learning paradigms that are inherent in all neural networks were also deliberated on.

Neural networks need to be trained first with sufficient amount of data in order to be used for modeling and forecasting. The neural networks thus learned to recognize some hidden trends in the data. The learning algorithm, called the delta rule, which is often utilized by the most common class of neural networks, was discussed. Backpropagation of errors using gradient descent algorithm in order to move the neural networks toward the global minimum was discussed. A detailed derivation of the formulae was also offered. Lastly, some successful implementation of MLP on various problems was also discussed.

2.1 The Main Concept of the Neural Networks Theory

2.1.1 The Basics of Neural Networks

According to the DARPA Neural Network Study (1988, AFCEA International Press, p. 60):

> a neural network is a system composed of many simple processing elements operating in parallel whose function is determined by network structure, connection strengths, and the processing performed at computing elements or nodes.

According to Haykin, S. (1994), Neural Networks: A Comprehensive Foundation, NY: Macmillan, p. 2:

> A neural network is a massively parallel distributed processor that has a natural propensity for storing experiential knowledge and making it available for use. It resembles the brain in two respects:
>
> 1. Knowledge is acquired by the network through a learning process.
> 2. Interneuron connection strengths known as synaptic weights are used to store the knowledge.

According to Nigrin, A. (1993), Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press:

> A neural network is a circuit composed of a very large number of simple processing elements that are neurally based. Each element operates only on local information. Furthermore each element operates asynchronously; thus there is no overall system clock.

According to Zurada, J.M. (1992), Introduction to Artificial Neural Systems, Boston: PWS Publishing Company:

> Artificial neural systems, or neural networks, are physical cellular systems which can acquire, store, and utilize experiential knowledge.

Artificial Neural Networks (ANN) have been applied to an increasing number of real-world problems of considerable complexity. Their most important advantage is in solving problems that are too complex for conventional technologies -- problems that do not have an algorithmic solution or for which an algorithmic solution is too complex to be found. In general, because of their abstraction from the biological brain, ANN are well suited to problems that people are good at solving, but for which computers are not. These problems include pattern recognition and forecasting (which requires the recognition of trends in data).

According to Eberhart and Dobbins (1990), neural network is an analysis tool that is modeled after the massively parallel structure of the brain. It simulates a highly interconnected, parallel computational structure with many relatively simple individual processing elements or nodes. Gurney (1996) stated that a neural network is an interconnected assembly of simple processing elements, units or nodes, whose functionality is loosely based on the animal neurone. The processing ability of the network is stored in the inter-unit connection strengths, or weights, obtained by a process of adaptation to, or learning from, a set of training patterns.

Neural networks, with their remarkable ability to derive meaning from complicated or imprecise data, can be used to extract patterns and detect trends that are too complex to be noticed by either humans or other computer techniques. A trained neural network can be thought of as an "expert" in the category of information it has been given to analyze. This expert can then be used to provide projections/predictions given new situations of interest and answer "what if" questions.

At their most fundamental level, neural networks are simply a new way of analyzing your data. What makes them so useful is their ability to learn complex patterns and trends in your data, an ability that is unique to neural networks. By using current data to build a model of the relationships between inputs and outputs, we can readily predict new output values based on previously unseen inputs.

2.1.2 **Learning in Artificial Neural Networks**

Artificial Neural Networks are computational systems whose architecture and operation are inspired from our knowledge about biological neural cells (neurons) in the brain. ANN can be described either as mathematical and computational models for non-linear function approximation, data classification, clustering and non-parametric regression or as simulations of the behavior of collections of model biological neurons. These are not simulations of real neurons in the sense that they do not model the biology, chemistry, or physics of a real neuron. They do, however, model several aspects of the information combining and pattern recognition behavior of real neurons in a simple yet meaningful way. Neural networks modeling has shown incredible capability for emulation, analysis, prediction, and association.

ANN can be used in a variety of powerful ways:

- to learn and reproduce rules or operations from given examples;
- to analyze and generalize from sample facts and make predictions from these;
- to memorize characteristics and features of given data and to match or make associations from new data to the old data.

ANN are able to solve difficult problems in a way that resembles human intelligence. What is unique about neural networks is their ability to learn by example. Traditional artificial intelligence (AI) solutions rely on symbolic processing of the data, an approach that requires a priori human knowledge about the problem. Neural networks techniques have an advantage over statistical methods of data classification because they are distribution-free and require no a priori knowledge about the statistical distributions of the classes in the data sources in order to classify them. Unlike these two approaches, ANN are able to solve problems without any a priori assumptions. As long as enough data is available, a neural network will extract any regularity and form a solution.

As ANN are models of biological neural structures, the starting point for any kind of neural network analysis is a model neuron whose behavior follows closely our understanding of how real neurons work. This model neuron is shown in Fig. 2.1.

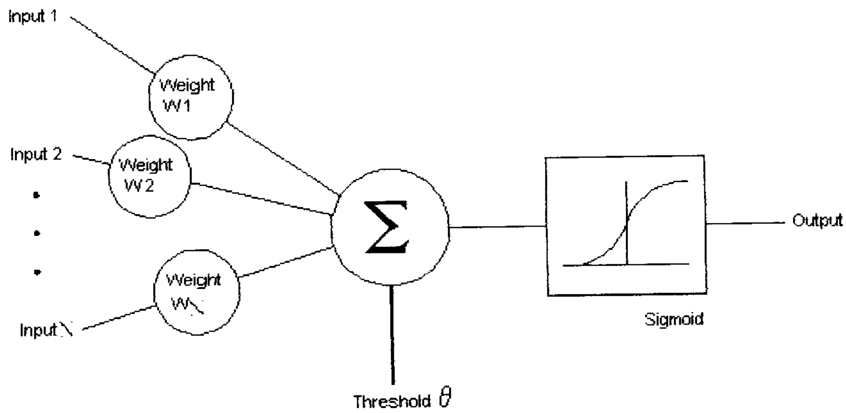


Figure 2.1: A Neuron Model

The neuron has N input lines and a single output. Each input signal is weighted; that is, it is multiplied with the weight value of the corresponding input line (by analogy to the synaptic strength of the connections of real neurons). The neuron will combine these weighted inputs by forming their sum and, with reference to a threshold value and activation function, it will determine its output. In mathematical terms, we may describe the neuron by writing the following pair of equations:

$$u = \sum_{i=1}^{N} w_i x_i \tag{2.1}$$

and

$$y = f(u - \theta) \tag{2.2}$$

where $x_1, x_2, \ldots, x_N$ are the input signals, $w_1, w_2, \ldots, w_N$ are the synaptic weights, u is the activation potential of the neuron, θ is the threshold, y is the output signal of the neuron, and f (.) is the activation function.

For notational convenience, the above equations may be reformulated by letting $w_0 = \theta$ and setting $x_0 = -1$. Then

$$\sum_{i=1}^{N} w_i x_i - \theta = \sum_{i=0}^{N} w_i x_i \tag{2.3}$$

and

$$y = f\left(\sum_{i=0}^{N} w_i x_i\right) \tag{2.4}$$

The combination of a fixed input $x_0 = -1$ and of an extra input weight $w_0 = \theta$ accounts for what is known as a bias input. Note that the new notation has augmented any input vector $\mathbf{x} \in \Re^N$ to the vector (-1, $\mathbf{x}$) $\in \Re^{N+1}$, and also the weight vector $\mathbf{w} \in \Re^N$ of the neuron, to the vector ($w_0, \mathbf{w}$) $\in \Re^{N+1}$.

The activation function, denoted by f (.), defines the output of the neuron in terms of the activity level at its input. The most common form of activation function used in the construction of ANNs is the sigmoid function. An example of the sigmoid is the logistic function, defined by

$$f(u) = \frac{1}{1+\exp(-au)} \tag{2.5}$$

where a is the slope parameter of the sigmoid function. By varying the parameter a, we can obtain sigmoid functions of different slopes. In the limit, as the slope parameter approaches infinity, the sigmoid function becomes simply a threshold function. The threshold function however, can take only the values 0 or 1, whereas a sigmoid function assumes a continuous range of values from 0 to 1. Also the sigmoid function is differentiable, whereas the threshold function is not. Differentiability is an important feature of neural network theory since it has a fundamental role in the learning process in ANN.

The first model of a neuron was proposed by McCulloch and Pitts (1943) when they described a logical calculus of neural networks. In this model the activation function used was the threshold function. The McCulloch-Pitts neuron models, connected up in a simple fashion (forming a single layer), were given the name "perceptrons" by Frank Rosenblatt. It represented his attempt to

> illustrate some of the fundamental properties of intelligent systems in general, without becoming too deeply enmeshed in the special, and frequently unknown, conditions which hold for particular biological organisms. (1985, p. 387)

He described the properties of these neurons, but more importantly he presented a method by which the perceptrons could be trained in order to perform simple patterns recognition tasks. He also provided a theorem called the perceptron convergence theory which guarantees that if the learning task is linearly separable (that is, if the data classes can be separated by a straight line in input space) then the perceptron will yield a solution in a finite number of steps.

Perceptrons, however, are unable to solve problems that are not linearly separable. It was the pointing of this limitation of the perceptrons by Minsky and Papert (1969) in their famous book "Perceptrons" (using elegant mathematical analysis to demonstrate that there are fundamental limits on what one-layer perceptrons can compute) and their pessimism about the prospects of discovering efficient algorithms for the training of MLP (MLP can solve non-linearly separable problems) which lead to the recline of the subject of neural computing for more than a decade.

The development, however, of the backpropagation algorithm by Rumelhart, Hinton and Williams (1986) and the subsequent publication of the book "Parallel Distributed Processing: Explorations in the Microstructures of Cognition" by Rumelhart and McClelland (1986) answered Minsky and Papert's (1969) challenge (in the sense that it was proved that there can indeed exist algorithms for the training of MLP) and that resulted in the resurgence of interest in neural computing.

There is a fair understanding of how an individual neuron works. However there is still a great deal of research and mostly conjecture regarding the way real neurons organize themselves and the mechanisms used by arrays of neurons to adapt their behavior to external stimuli. There are a large number of experimental ANN structures currently in use reflecting this state of continuing research. Among the many interesting properties of all these structures, the property that is of primary significance is the ability of the networks to learn from their environment, and to improve their performance through learning. ANN learns about their environment through an iterative process of adjustments applied to their free parameters, which are the synaptic weights and thresholds. The type of learning is determined by the manner in which the parameter changes take place.

There are three basic types of learning paradigms:

1. supervised learning,
2. reinforcement learning, and
3. self-organized (unsupervised) learning.

As its name implies supervised learning is performed under the supervision of an external "teacher". The teacher provides the network with a desired or target response for any input vector. The actual response of the network to each input vector is then compared by the teacher with the desired response for that vector, and the network parameters are adjusted in accordance with an error signal which is defined as the difference between the desired response and the actual response. The adjustment is carried out iteratively in a step-by-step fashion with the aim of eventually making the error signal for all input vectors as small as possible. When this has been achieved then the network is believed to have built internal representations of the data set by detecting its basic features, and hence, to be able to deal with data that has not encountered during the learning process, that is, it can generalize its "knowledge".

Supervised learning is by far the most widely used learning technique in ANN because of the development of the backpropagation algorithm, which allows for the training of multilayer ANN.

MLP is a modeling and forecasting tool that uses a neural network to model your data. It can be used to classify patterns or to predict values from your data. It can be used to produce good models that accurately represent nonlinearities in your data, as is the case with the data to be collected for this project. A MLP consists of an input layer, an output layer, and an additional layer of neurons between the input and output layer, known as the hidden layer. This hidden layer vastly increases its learning power. It uses a transfer, or activation function to modify the input to a neuron, usually the sigmoid function.

Training proceeds in the following way. First, the weights and biases in the network are initialized, usually to small random values. A training pattern is then applied to the input units and the activation of neurons in the first hidden layer are calculated. The outputs produced by these neurons via the transfer function are then fed to neurons in the following layer. This forward pass process is repeated at each layer until an output signal from neurons in the output layer is obtained.

The difference between the actual and desired output values is measured, and the network model connection strengths are changed so that the outputs produced become closer to the desired outputs. This is achieved by a backward pass during which connection changes are propagated back through the network, starting with the connections to the output layer and ending with those to the input layer.

The basic method for adapting the connections, the learning rule, is as follows. If the output produced by the network is correct, the connections from the output neurons to all input neurons are unchanged. If the network output is larger than the desired output at any node, the connections are decreased. If the outputs are smaller than desired, the connections are increased. Once the best solution has been found, the weights are fixed, and no further training takes place.

However, care must be taken into consideration when selecting appropriate number of hidden units to achieve a proper solution. Fadzilah Siraj, in "A Survey of Pruning Algorithms in Neural Networks" (1994), described the pruning algorithms that can be employed for eliminating or pruning hidden units in multilayer neural networks after training or during training process. Pruning algorithms can be classified into three categories; train a network that is larger than required and then prune the network, add or remove the hidden units dynamically during the training progress, and use the information criteria to determine the optimal size of network.

Neural networks are typically organized in layers. Layers are made up of a number of interconnected 'nodes' that contain an 'activation function'. Patterns are presented to the network via the 'input layer', which communicates to one or more 'hidden layers' where the actual processing is done via a system of weighted 'connections'. The hidden layers then link to an 'output layer' where the answer is output as shown in Figure 2.2.

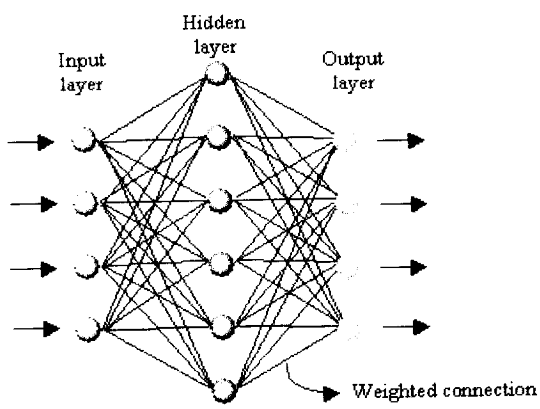


Figure 2.2: A Typical Neural Networks Organized in Layers

Most ANN contained some form of 'learning rule' that modified the weights of the connections according to the input patterns that it was presented with. In a sense, ANN learned by example, as do their biological counterparts; a child learns to recognize people from examples of people.

Although there are many different kinds of learning rules used by neural networks, this project is concerned only with one; the Delta Rule. The delta rule is often utilized by the most common class of ANN called 'backpropagational neural networks' (BPNN). Backpropagation is an abbreviation for the backward propagation of error.

With the delta rule, as with other types of backpropagation, 'learning' is a supervised process that occurs with each cycle or 'epoch' (i.e. each time the network is presented with a new input pattern) through a forward activation flow of outputs, and the backwards error propagation of weight adjustments.

More simply, when a neural network is initially presented with a pattern it makes a random 'guess' as to what it might be. It then sees how far its answer was from the actual one and makes an appropriate adjustment to its connection weights. More graphically, the process looks something like this (Figure 2.3):

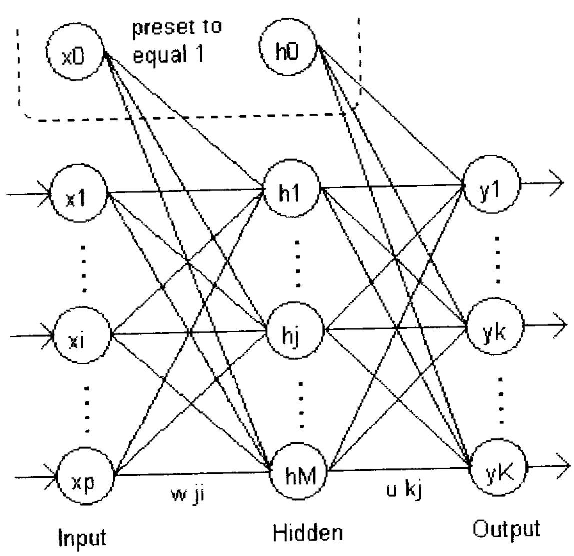


Figure 2.3: Connection Weights between Layers

Note that within each hidden layer node there is a sigmoidal activation function that polarizes network activity and helps stabilize it.

Backpropagation performs a gradient descent within the solution's vector space towards a 'global minimum' along the steepest vector of the error surface. The global minimum is that theoretical solution with the lowest possible error. The error surface itself is a hyperparaboloid but is seldom 'smooth' as is depicted in the graphic below. Indeed, in most problems, the solution space is quite irregular with numerous 'pits' and 'hills', which may cause the network to settle down in a 'local minimum' which, is not the best overall solution.

Since the nature of the error space can not be known a priori, neural network analysis often requires a large number of individual runs to determine the best solution. Most learning rules have built-in mathematical terms to assist in this process, which control the 'speed' and the 'momentum' of the learning. The speed of learning is actually the rate of convergence between the current solution and the global minimum. (Figure 2.4) Momentum helps the network to overcome obstacles (local minima) in the error surface and settle down at or near the global minimum.

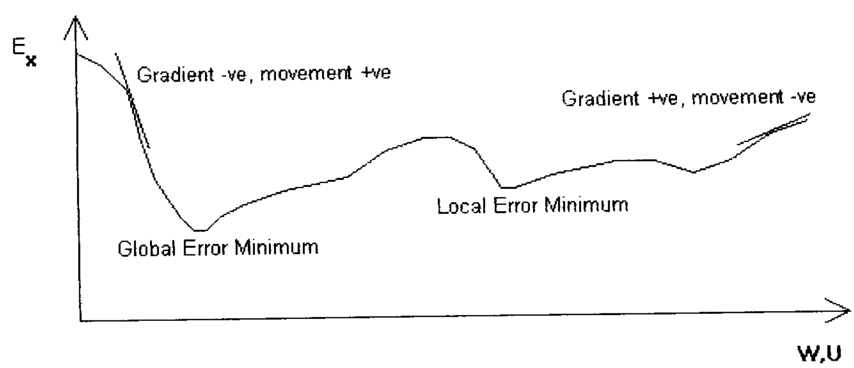


Figure 2.4: Convergence towards Global or Local Minimum

Once a neural network is 'trained' to a satisfactory level it may be used as an analytical tool on other data. To do this, the user no longer specifies any training runs and instead allows the network to work in forward propagation mode only. New inputs are presented to the input pattern where they filter into and are processed by the middle layers as though training were taking place, however, at this point the output is retained and no backpropagation occurs. The output of a forward propagation run is the predicted model for the data that can then be used for further analysis and interpretation.

However, it is also possible to over-train a neural network, which means that the network has been trained exactly to respond to only one type of input. This is much like rote memorization. If this should happen then learning can no longer occur and the network is referred to as having been "overfitted". In real-world applications this situation is not very useful since one would need a separate overfitted network for each new kind of input.

2.1.3 General Multilayer Perceptrons Networks

MLP are feedforward neural networks trained with the standard backpropagation algorithm. They are supervised networks that require a desired response to be trained. They learn how to transform input data into a desired response. Because of this, they are widely used for pattern classification. With one or two hidden layers, they can approximate virtually any input-output map.

MLP neural networks are composed of configurations of simple perceptrons in a hierarchical structure forming a feedforward topology. They have one or more hidden layers of perceptrons between the input and output layers. It is permissible to have any prior layer nodes connected to subsequent layer nodes via a corresponding set of weights. Different learning algorithms can be used for MLP, but the most common ones have been the delta and backpropagation-of-error algorithms. These algorithms do work fairly well but they tend to be slow.

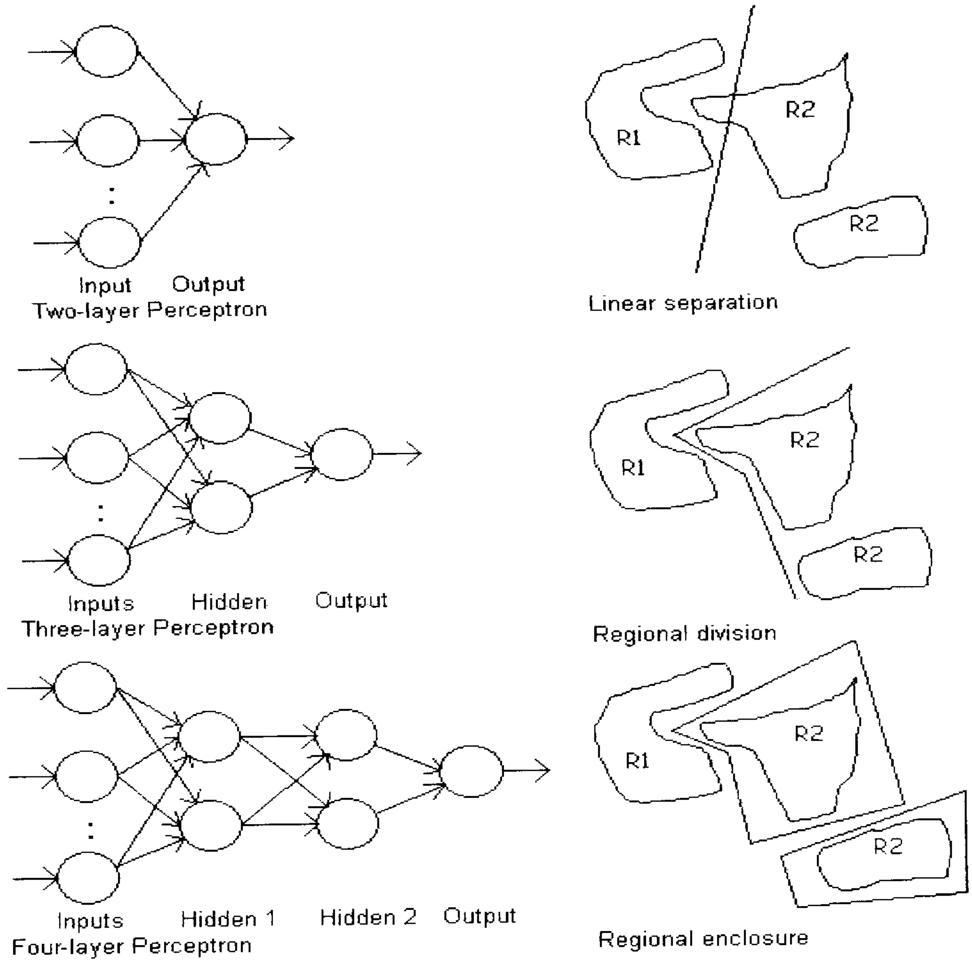


Figure 2.5: Classification Ability of Perceptrons

The diagrams in Figure 2.5 show simple examples of the general classification ability of perceptrons configured in various ways. A two-layer perceptron network can only solve linearly separable problems, a three-layer network can divide complex regions and a four-layer network can fully enclose complex regions.

The general model, with one hidden layer is sufficient to represent the MLP. By adding a continuously differentiable Processing Element PE function, such as a sigmoid, it is possible, at least in theory, for a network with one or more hidden layers to perform practically any nonlinear mapping to any desired degree of accuracy.

Error correction learning is easy to apply to a network without any hidden layers.

If, the input signal vector is $x = [x_1, x_2, \ldots\ldots, x_p]^T$

and the error signal vector is $e = (d - y)$,

where $d = [d_1, d_2, \ldots\ldots, d_k]^T$

is the desired output vector response,

and $y = [y_1, y_2, \ldots\ldots, y_k]^T$

is a vector of the actual system response a simple Least Mean Square (LMS) weight learning law can be derived using gradient descent for a two-layer network (input and output layer) with no activation functions as follows:

$$W(k+1) = W(k) + \mu e.x(k)^T \tag{2.6}$$

where μ is a scalar gain or learning rate factor greater than zero.

A solution is attained when the error vector e has been minimized through the iterative process of presenting new input/output pairs to the learning algorithm and making the weight adjustments.

To apply the LMS learning algorithm the error function for a given input vector x and K output nodes is defined as:

$$E_x = \frac{1}{2}\sum_{k=1}^{K}(d_k - y_k)^2 \tag{2.7}$$

By moving in the opposite direction of the gradient of this error function, with respect to the weights, the optimal solution can be achieved provided the gain μ is made sufficiently small. The error function relates only to a single input vector x. It has been shown by Widrow and Hoff (1960, pp.96-104), as seen in equation

(2.6), that moving in the opposite direction of the gradient for each vector, when taken in aggregate, still achieves convergence.

Beyond a 2-layer network, error correction learning is more difficult to achieve because the amount of error that each hidden node contributes to the output layer must be computed. This problem was not solved until it was realized that a continuously differentiable PE function would allow the chain rule of partial differentials to be used to calculate the weight changes for any weight in the network. This process is also referred to as the backpropagation-of-errors or simply backpropagation.

The MLP model with one hidden layer can be expressed as a nonlinear multivariate function, $F(x, W, U)$ of the input vector x, the weight matrix W of weights between the input and hidden layers, the weight matrix U of the weights between the hidden layer and the output layer, and the activation functions:

$$y = F(x, W, U) \tag{2.8}$$

The activation functions associated with the hidden layers are sigmoids whereas the activation functions associated with the outputs can be sigmoidal for classification applications, or linear for function mapping applications. For classification problems it is common to assign one output node for each class, with an output of 0.9 representing class selection and 0.1 class rejection.

The general gradient descent weight learning laws for weights W and U can be defined as:

$$W(k+1) = W(k) - \mu \frac{\partial E_x}{\partial W} \tag{2.9}$$

$$U(k=1) = U(k) - \eta \frac{\partial E_x}{\partial U} \qquad (2.10)$$

where μ and η are positive valued scalar gain or learning rate constants, less than 1.

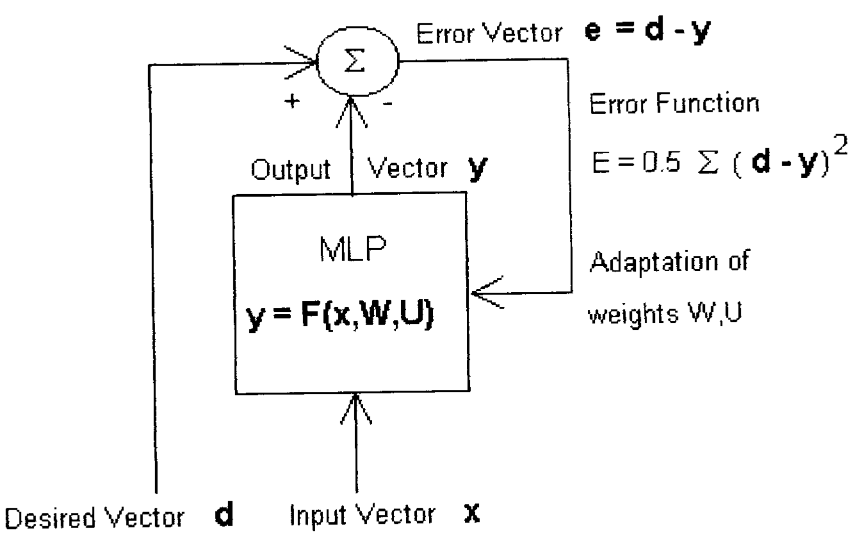


Figure 2.6: Multi-layer Perceptron Block Diagram

Backpropagation-of-error learning involves minimizing the error function given by equation (2.7) by incremental adjustments of all the weights W and $\boldsymbol{U}$ according to some learning law such as gradient descent, as each vector pair d and x is presented to it in turn. Minimization of the error needs special attention, as the minimum could be either a local minimum or a global minimum as illustrated in Figure 2.4.

2.1.4 A Detailed Three-Layer MLP Model

The three-layer feedforward MLP has an input, one hidden layer and an output layer. The input layer is only a "fan-out" layer, where the input vector is distributed to all the hidden layer PE (neurons or nodes). There is no real processing done in this layer. The hidden layer is the key to the operation of the MLP. Each of the hidden nodes is a single PE that implements its own decision surface.

The output layer is a set of decision surfaces in which each of its PE has decided what part of the decision space the input vector lies. The role of the output layer is essentially to combine all of the "votes" of the hidden layer PE and decide upon the overall classification of the vector. The nonlinearity provided by the activation functions of the hidden and output PE allows this network to solve complex problems that are not linearly separable. This is done by forming complex decision surfaces by a nonlinear combination of the hidden layer's decision surfaces.

Figure 2.6 represents a three-layer feedforward MLP model block diagram. After training the feedforward equations relating the inputs to the outputs are described by the following matrix equation (2.11):

$$y = \mathbf{f}(U\,\mathbf{f}\,((W_x))) \qquad (2.11)$$

where

$x = [1, x_1, x_2, \ldots, x_i, \ldots, x_p]^T$ input vector $((p + 1) \times 1)$

W = matrix of weights w_{ji} between input and hidden nodes $((M + 1) \times (p + 1))$

U = matrix of weights u_{kj} between hidden and output nodes $(K \times (M + 1))$

y = output vector $(M \times 1)$

$f(.)$ = multivariate activation function

p = number of real input nodes

M = number of real hidden nodes

K = number of output nodes

Equation (2.11) can be expressed in detail by equations (2.12), (2.13) and (2.14)

$$h_j = f_j\left(\sum_{i=0}^{p} w_{ji} x_i\right),$$

as follows:

$$\text{for } j = 1, 2, \ldots, M \tag{2.12}$$

$$y_k = f_k\left(\sum_{j=0}^{M} u_{kj} h_j\right),$$

$$\text{for k} = 1, 2, \ldots, \text{K} \tag{2.13}$$

for activation function

$$f(z) = f_j(z) = \frac{1}{1+e^{-z}} \tag{2.14}$$

The derivative of the activation function is a function of itself and is defined by equation (2.15). This useful property will be used later in the MLP learning algorithm.

$$f' = \frac{df(z)}{dz} = \frac{e^{-z}}{\left(1+e^{-z}\right)^2} = \frac{1}{1+e^{-z}}\left(1 - \frac{1}{1+e^{-z}}\right) = f(1-f) \tag{2.15}$$

We can see that the outputs y_k are a function of hidden outputs h_j and the weights u_{kj} between the hidden layer and the output. The outputs h_j are a function of the inputs x_i and the weights between the inputs and the hidden layer. Note that the weights between the inputs and the internal bias node h_o are zero and the bias inputs x_o and h_0 are set to equal 1.

2.1.5 MLP Backpropagation-of-error Learning

The MLP stores its knowledge in the weights. The problem is to adjust these weights in a way that will produce the required knowledge and solutions to our problems. Because the type of classification problems we are generally interested in are too complex to solve a priori by analytic techniques we need to develop a training algorithm which is driven by example data.

The hope is that if we have adequate features, number of PE and sufficient representative training data samples we can slowly adjust the weights through training in such a way as to end up with a set of network weights which will give a satisfactory classification performance for other inputs which the network has not seen during training. We can achieve this optimization most effectively by adjusting the weights to minimize the Mean Squared Error (MSE) of the network outputs compared with desired responses. This can be very time consuming if it is necessary to compute the MSE of all the training pairs before the weights can be incrementally adjusted once.

Alternatively, it is possible to use backpropagation-of-error learning which is also based on the gradient descent optimization technique. The main idea behind backpropagation-of-error learning is to adjust the weights a little each time as a new random training input/output vector pair is presented to the network. When using gradient descent, the local gradient of the error function, equation (2.7), is computed and the weights are then adjusted in the opposite direction to the gradient. This is the direction that on the whole makes the overall network error smaller. The main problem with gradient descent optimization is that it can be prone to converging to a local minimum instead of the global minimum. There are a number of techniques including the so-called "simulated annealing" that have been developed to try to solve this problem. A more comprehensive details on the backpropagation-of-error algorithm can be referred to the works of the co-inventors, Parker (1985) and Rumelhart (1986, pp. 533-536).

2.1.6 Derivation of Backpropagation-of-error Learning

After training we require that the MSE be minimized for the whole training set of input/output vector pairs. To achieve this we have to adjust the two sets of network weights, the output layer weights, u_{kj} and the hidden layer weights, w_{ji}. We will need to calculate the gradient of the error in the whole weight space. To do this we can use partial derivatives and the chain rule to calculate the contribution that each of the weights makes on the total error as follows.

2.1.7 Change in Error due to Output Layer Weights

The partial derivative of the error with respect to the output layer weights is:

$$\frac{\partial E_x}{\partial u_{kj}} = \frac{\partial E_x}{\partial y_k} \cdot \frac{\partial y_k}{\partial u_{kj}} \tag{2.16}$$

Equation (2.16) is made up from the partial derivative of the error function multiplied by the derivative of the output generating function. If we substitute the error function equation (2.7) into equation (2.16) we get equations (2.17) to (2.20):

$$\frac{\partial E_x}{\partial u_{kj}} = \frac{\partial}{\partial y_k}\left[\frac{1}{2}\sum_{a=1}^{K}(d_a - y_a)^2\right] \cdot \frac{\partial}{\partial u_{kj}}\left[f_k(\sum_{b=0}^{M}(u_{kb}.h_b))\right] \tag{2.17}$$

$$\frac{\partial E_x}{\partial u_{kj}} = (y_k - d_k).f_k'\left(\sum_{b=0}^{M} u_{kb}.h_b\right).h_j$$

(2.18)

$$\frac{\partial E_x}{\partial u_{kj}} = \delta y_k.h_j$$

(2.19)

where

$$\delta y_k = (y_k - d_k).f_k'\left(\sum_{b=0}^{M} u_{kb}.h_b\right)$$

(2.20)

represents the backpropagating error related to the hidden layer.

2.1.8 Change in Error due to Hidden Layer Weights

The calculation of the change in error as a function of the hidden layer weights is more difficult because there is no way of getting "desired outputs" for the hidden layer PE. We only know what the network outputs should be. The partial derivative is similar to before but a little more complex.

$$\frac{\partial E_x}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}}\left[\frac{1}{2}\sum_{a=1}^{K}(d_a - y_a)^2\right]$$

(2.21)

$$\frac{\partial E_x}{\partial w_{ji}} = \sum_{a=1}^{K} \frac{\partial}{\partial w_{ji}} \left[\frac{1}{2} (d_a - y_a)^2 \right]$$

(2.22)

$$\frac{\partial E_x}{\partial w_{ji}} = \sum_{a=1}^{K} \left[\frac{\partial}{\partial y_a} \left(\frac{1}{2} (d_a - y_a)^2 \right) . \frac{\partial y_a}{\partial h_j} . \frac{\partial h_j}{\partial w_{ji}} \right]$$

(2.23)

$$\frac{\partial E_x}{\partial w_{ji}} = \left[\sum_{a=1}^{K} (y_a - d_a) . f_k^{'} \left(\sum_{b=0}^{M} u_k b . h_b \right) . u_{aj} \right] . f_j^{'} \left(\sum_{b=0}^{p} w_{jb} . x_b \right) . x_i$$

(2.24)

$$\frac{\partial E_x}{\partial w_{ji}} = \delta h_j . x_i$$

(2.25)

$$\delta h_j = \left[\sum_{a=1}^{K} (y_a - d_a) . f_k^{'} \left(\sum_{b=0}^{M} u_{kb} . h_b \right) . u_{aj} \right] . f_j^{'} \left(\sum_{b=0}^{p} w_{jb} . x_b \right)$$

(2.26)

which represents the backpropagation of the error from the output layer to the hidden layer.

2.1.9 The Weight Adjustments

We can now see that in order to minimize the error, we should adjust all the weights in the opposite direction to the error gradient each time we present a training input/output vector pair to the network as follows:

$$\Delta u_{kj} = -\eta.\frac{\partial E_x}{\partial u_{kj}} = -\eta.\delta y_k.h_j \tag{2.27}$$

$$u_{kj}^{new} = u_{kj}^{old} + \delta u_{kj} \tag{2.28}$$

$$\delta w_{ji} = -\mu.\frac{\partial E_x}{\partial w_{ji}} = -\mu.\delta h_j.x_i \tag{2.29}$$

$$w_{ji}^{new} = w_{ji}^{old} + \Delta w_{ji} \tag{2.30}$$

where μ and η are positive valued scalar gain or learning rate constants.

The learning rate is controlled by the scalar constants μ and η. These should be relatively small, ie μ and $\eta < 1$. If they are too small the rate of convergence is slow, but if they are too large it may be difficult to converge once in the vicinity of a minimum since the estimate of the gradient is only valid locally. If we move too far then we may begin to move in a direction not reflected by the gradient.

The ideal learning strategy may be to use relatively high values to start with and then reduce them as the training progresses. When we only have a finite training vector set it is advisable to continually select the individual training vector input/output pairs at random from the set rather than sequence through the set. The training may require many 100 000's or even 1 000 000 's of these iterations, especially for very complex problems.

For these equations to work we need an activation function which is differentiable and if possible one whose derivative is easy to compute. The sigmoid function of equation (2.14) is a suitable function because not only is it continuously differentiable, its derivative is a simple function of itself, as seen in equation (2.15). Therefore, we can rewrite the weight adjustment equations (2.27) and (2.29) simply as equations (2.31) and (2.33) respectively:

$$\begin{aligned}\Delta u_{kj} &= -\eta.\delta y_k.h_j \\ &= -\eta[(y_k - d_k).f_k.(1 - f_k)]h_j \\ &= -\eta.[(y_k - d_k).y_k.(1 - y_k)]h_j\end{aligned}$$

$$\text{for } k = 1, \ldots, K \text{ and } j = 0, \ldots, M \tag{2.31}$$

where:

$$f_k = f_k\left(\sum_{b=0}^{M} u_{kb}.h_b\right) = \frac{1}{1 + \exp\left(-\sum_{b=0}^{M} u_{kb}.h_b\right)} = y_k \tag{2.32}$$

$$\Delta w_{ji} = -\mu.\delta h_j.x_j$$

$$= -\mu.\left[\left(\sum_{a=1}^{K}(y_a - d_a).f_k'\left(\sum_{b=0}^{M} u_{kb}.h_b\right).u_{aj}\right).f_j.(1-f_j)\right].x_j$$

$$= -\mu\left[\left(\sum_{a=1}^{K}\delta y_a.u_{aj}\right).f_j.(1-f_j)\right].x_i$$

$$= -\mu\left[\left(\sum_{a=1}^{K}\delta y_a.u_{aj}\right).h_j.(1-h_j)\right].x_i$$

for $j = 1, \ldots, M$ and $i = 0, \ldots, p$ (2.33)

where:

$$f_j = f_j\left(\sum_{b=0}^{p} w_{jb}.x_b\right) = \frac{1}{1+\exp\left(-\sum_{b=0}^{p} w_{jb}.x_b\right)} = h_j$$

(2.34)

Equations (2.31) and (2.33) are now in terms of the actual input, node output and desired values and can therefore be computed by simple arithmetic.

2.1.10 Additional Momentum Factor

When the network weights approach a minimum solution, the gradient becomes small and the step size diminishes too, giving very slow convergence. If a so-called momentum factor is added to the weight update equations the weights can be updated with some component of past updates. This can reduce the decay in learning updates and cause the learning to proceed through the weight space in a fairly constant direction. The momentum factor determines the proportion of the old weight change that is added into the new weight change while the learning rate determines the amount of weight change (Koekkoek, 1996).

The benefits of this, apart from faster convergence to the minimum, is that it may even be possible to sometimes escape a local minimum if there is enough momentum to travel through it and over the following hill.

Adding the momentum factor to the gradient descent learning equations (2.9) and (2.10) modifies to equations (2.35) and (2.36) respectively.

$$W(k+1) = W(k) - \mu \frac{\partial E_x}{\partial W} + \alpha(W(k) - W(k-1)) \tag{2.35}$$

$$U(k+1) = U(k) - \eta \frac{\partial E_x}{\partial U} + \beta(U(k) - U(k-1)) \tag{2.36}$$

where μ, η, α and β are positive valued scalar gain or learning rate constants, all less than 1.

When the gradient has the same algebraic sign on consecutive iterations the weight change grows in magnitude. Thus momentum tends to accelerate descent in steady downhill directions. When the gradient has alternating algebraic signs on consecutive iterations the weight changes become smaller, thus stabilizing the learning by preventing oscillations.

The following procedures, offered by Freeman and Skapura (1992), summarizes the order in which they would be used during training for a single training-vector pair:

1. Apply the input vector to the input units.
2. Calculate the net-input values to the hidden layer units.
3. Calculate the outputs from the hidden layer.
4. Move to the output layer. Calculate the net-input values to each unit.
5. Calculate the outputs.
6. Calculate the error terms for the output units.
7. Calculate the error terms for the hidden units. (Notice that the error terms on the hidden units are calculated before the connection weights to the output layer units have been updated.)
8. Update weights on the output layer.
9. Update weights on the output layer. Be sure to calculate the error term.

2.2 Successful Implementations Using MLP

Caren Marzban, E. DeWayne Mitchell and Gregory J. Stumpf (1996) have developed a successful neural network for tornado diagnosis using MLP. This MLP model has been designed to classify the two types of circulations – nontornadic and tronadic – based on various attributes of the circulations. This study showed the ability of MLP to offer an accurate model for tornado diagnosis, thus giving confidence to its use in other problems.

A backpropagation neural network, trained to make correct diagnosis of thyroid disease on the basis of signs and symptoms, was compared to human physicians. The result showed that the neural network was able to make the correct diagnosis in all cases whereas the human physicians were unable to make the correct diagnosis (Hong, Z and Frank C.L, 1998).

The most commonly and established neural network model in medical diagnosis is Backpropagation. This can be seen in Salchenberg's paper on 'using neural networks to aid the diagnosis of breast implant rupture' by using Backpropagation and Radial Basis Function. Both showed better diagnosis than radiologist (Salchenberg.L et.al, 1997).

In another successful project, a neural network based on back propagation net was used for modeling non-linear chemical systems (Bhat, Minderman, McAvoy and Wang, 1990). They had used back propagation successfully on a number of problems typical of those found in the chemical/petroleum industry, including sensor interpretation, dynamic modeling, and in learning how to design distillation control systems.

Le Cun, Boser, Denker, Henderson, Howard, Hubbard and Jackel (1990) wrote about how back propagation networks could be applied to real image-recognition problems without a large, complex pre-processing stage requiring detailed engineering. This application was designed to handwritten digit recognition. The input of the network consisted of normalized images of isolated digits. The method has 1% error rate and about a 9% reject rate on zipcode digits provided by the US Postal Service.

Vincent, Myers and Hutchinson (1992) produced a system, based on the use of neural network feature detectors, to robustly locate and track features in digital image sequences. They had concentrated on the location of eyes and mouths in human head-and-shoulders images. MLP was used in this experiment.

Woodland (1992) discussed the used of MLP as a classifier in his speech recognition system. A number of MLPs were trained on both the speaker independent and multiple speaker datasets. A dynamic time wrapping (DTW) system was also trained and tested. He found that the MLP recognizer was more accurate than the RTW system. Similar systems were also discussed in detail by Mason and Andrews on Speech and speaker recognition, Smyth on segmental sub-word unit classification, and Ainsworth and Warren on text-to-speech synthesis systems.

In spite of their widespread use for data modeling in economics, physics, computer science and pattern recognition, neural networks are almost unknown in the educational sciences community. Tirri, H., Silander, T. and Tirri, K. (1997) concurred that this is perhaps partly caused by the unfamiliar terminology associated with neural networks due to their origin from cognitive and neurosciences, partly by the lack of demonstrations of the applicability of the methods for educational data.

CHAPTER 3
METHODOLOGY

3.0 Overview of Chapter

This chapter outlined the methodology used in the pre-collection of data, where the study area and data collection procedures were discussed. Examination marks of subjects chosen as input variables will also be elaborated on. The research procedure, starting with the various applications to the relevant departments for approvals to proceed with the project, and the post-collection exercise where steps were taken to prepare the collected raw data to be ready to be submitted to the neural networks to be trained were also discussed.

Statistical testing was also done to determine to what extend the collected data were normally distribution. A brief discussion was also offered to attempt to explain why some of the input variables were slightly skewed. Tests were also done to determine the correlation between all input variables with the target variable. To assist the reader in the usage of the Neuron Connection Software Package, a detailed user manual is offered in Appendix A.

3.1 The Study Area and Data Collection Procedures

The identification of suitable variables/attributes to be accepted as input variables into the neural network had to be pre-determined first. Statistical testing was then performed to test the correlation between these inputs with the targeted output. This was to ensure that the neural network would be able to achieve an optimized model that would be able to predict to a high percentage bracket.

The input variables chosen were the student's Form 4 final examination, Form 5 mid-year examination and the SPM trial examination results. Actual marks scored, ranging from zero to 100, for subjects like Bahasa Malaysia, Bahasa Inggeris, Mathematics, Additional Mathematics, Physics, Chemistry, and Biology were collected. In addition to this, other input variables collected were the Aggregate scored, Grade achieved, whether urban or rural school, and whether boys or girls school.

The "target" input was the offer letter to Sixth Form Science Stream. This letter contained the list of successful students to be admitted into Sixth Form. A total of 1488 data samples were collected from 10 schools in the Kinta district of Perak. These schools, comprised of 6 girls schools and 4 boys schools, were carefully selected to represent as wide as possible the diverse spread of the different types of students in the district. These data consisted of records taken from year 1996 and 2000. Data samples were collected from three different batches of students. The first batch took their SPM in 1997, the second batch in 1998, and the third batch in 1999.

The input form prepared and used during this initial collection of data are as shown in Appendix B. Table 3.1 showed the input form with some sample data:

Table 3.1: Input form with sample data

Form 5 (Mid-Term Examinations)									Form 5 (Trial SPM)									Sixth Form
B.M.	B.I.	Math	Amath	Phy	Chem	Bio	Grd	Agg	B.M.	B.I.	Math	Amath	Phy	Chem	Bio	Grd	Agg	Offer
47	65	32	14	17	24	33	4	47	55	71	44	10	23	22	34	4	44	0
58	61	51	28	36	43	47	3	39	56	61	66	22	34	40	45	3	37	0
54	58	51	14	23	34	32	4	45	50	58	55	13	19	33	31	4	48	0
61	65	57	48	41	43	46	3	36	57	60	82	40	33	35	43	2	34	0
47	54	60	21	14	27	41	3	39	51	60	79	17	21	26	33	4	38	0
68	79	70	63	55	68	72	1	21	70	80	87	51	53	80	69	1	13	1
64	74	71	57	56	61	72	2	25	62	80	82	61	43	53	65	1	22	1
65	60	55	40	37	56	57	2	33	59	78	86	35	26	41	52	2	29	1
60	77	80	42	26	24	37	4	34	68	80	83	40	33	28	50	2	27	1
63	82	84	67	65	67	81	1	16	69	88	96	73	60	69	74	1	16	1

Raw marks scored during the Form 4 Final examination, the Form 5 Mid-Year examination and the Form 5 Trial examination for every subject were collected. Raw marks were considered the better choice as compared to subject grades because different schools used different criteria for grading. The Aggregates and Examination Grade were collected too. These records were usually kept as mark sheets in easily accessed files by the various schools. Most schools keep these in plain old trusted paper-based form but some schools keep them in computer files.

Care had to be taken when collecting these data samples, as the same student's marks must be followed from Form 4 to Form 5. Some schools do not change the students' class enrolment when they move on to Form 5. This ensured that they stayed in the same class and could be tracked easily. However, some schools re-streamed the classes when they moved on to Form 5 resulting in a total change in class enrolment. This resulted in a particular student becoming placed in a different class. In such cases, the researcher had to track the same student carefully and tediously.

Initial training and testing using Neural Connection Version 2 was then performed. Based on the MLP architecture, using Supervised topology and Steepest Gradient Descent training algorithm, a very high percentage of accuracy was achieved. By selecting different number of hidden layers and hidden nodes, different values for momentum and learning rate, the network was able to give a 90+% accuracy.

It is important that data samples used should be free of any bias as much as possible. In view of this, data samples collected should, as far as possible, consist of a nearly equal number of students scoring Grade 1, 2, 3 and 4. The number of data samples collected from urban schools and rural schools should also be roughly equal. The number of data samples from boys and girls schools should be about equal too. Extra care were taken into consideration during the exercise of gathering and processing data that should, as close as possible, represent the many different types of students in our school system.

Some constraints and limitations identified include:

- Some schools did not keep a complete record of examination results of some of the academic years needed for this project. For these schools, only available data will be collected and used.
- Some science students did not take Additional Mathematics and/or Physics for their SPM. There were no marks in these 2 columns. In processing these data, there were 2 options that the researchers could use. The first option was to leave it empty as it was. The second option, which was used, was to fill in these empty columns with the mean values.
- Some students were absent from their examination for certain subjects. Again there were not any marks scored for these subjects. As the numbers of fields with empty data for these particular students were rather large, the researcher opted to omit these data samples from the training procedures.

3.2 The Data's Statistics

Altogether, a total of 1488 samples were collected from 10 different schools of diverse background. The subjects chosen as inputs were Bahasa Malaysia, Bahasa Inggeris, Mathematics, Additional Mathematics, Physics, Chemistry and Biology. Raw marks were used as this offered a more accurate representation of the students' achievements as compared to using grades. This was because schools used different grading systems. In addition to this, the overall examination Grades and Aggregates were also selected as inputs. Aggregates were counted from the addition of subject grade from the 6 best subjects. Students were then accredited with either a Grade 1, 2, 3 or 4(for SAP and Fail), depending on the students meeting certain conditions.

The data were collected from the students' Form 4 Final examination, Form 5 Midyear examination and Form 5 Trial examination results. These data were taken from results between 1996 and 1999. The target in each case was whether the student was successful in getting an offer to study in Form 6. As this project focussed only on science students, the data were collected from science stream students only.

The table 3.2, 3.3 and 3.4 showed the summary of the statistics of each subject.

Table 3.2: Statistics of Form 4 Final Results

	BM F4	BI F4	MAT F4	AMAT F4	PHY F4	CHEM F4	BIO F4	GRD F4	AGG F4
NO OF CASES	1488	1488	1488	1488	1488	1488	1488	1488	1488
MINIMUM	0	0.1	0	0	0	0	0	0	0
MAXIMUM	0.83	0.95	1	0.99	0.98	0.97	0.97	1	1
RANGE	0.83	0.85	1	0.99	0.98	0.97	0.97	1	1
MEAN	0.559	0.611	0.637	0.399	0.488	0.522	0.578	0.381	0.444
VARIANCE	0.012	0.02	0.049	0.054	0.036	0.038	0.024	0.152	0.057
STANDARD DEV	0.107	0.143	0.22	0.232	0.191	0.196	0.156	0.39	0.239
STD. ERROR	0.003	0.004	0.006	0.006	0.005	0.005	0.004	0.01	0.006
SKEWNESS(G1)	-0.567	-0.27	-0.577	0.324	0.227	-0.051	-0.313	0.558	0.224
KURTOSIS(G2)	0.984	-0.382	-0.52	-0.804	-0.736	-0.724	0.117	-1.192	-0.874
SUM	831.96	909.84	947.35	593.59	726.62	776.9	860.49	566.333	659.979
C.V.	0.192	0.233	0.346	0.581	0.391	0.376	0.269	1.026	0.538
MEDIAN	0.565	0.62	0.67	0.4	0.47	0.53	0.58	0.333	0.438

Table 3.3 :Statistics of Form 5 Mid-Year Results

	BM F5M	BI F5M	MAT F5M	AMAT F5M	PHY F5M	CHEM F5M	BIO F5M	GRD F5M	AGG F5M
NO OF CASES	1488	1488	1488	1488	1488	1488	1488	1488	1488
MINIMUM	0.01	0.13	0.01	0	0.03	0.06	0.09	0	0
MAXIMUM	0.9	0.95	1	0.98	0.98	0.98	0.94	1	1
RANGE	0.89	0.82	0.99	0.98	0.95	0.92	0.85	1	1
MEAN	0.58	0.613	0.681	0.425	0.504	0.537	0.522	0.362	0.433
VARIANCE	0.011	0.018	0.043	0.062	0.039	0.041	0.03	0.162	0.062
STANDARD DEV	0.106	0.133	0.208	0.248	0.198	0.203	0.173	0.403	0.249
STD. ERROR	0.003	0.003	0.005	0.006	0.005	0.005	0.004	0.01	0.006
SKEWNESS(G1)	-0.832	-0.328	-0.761	0.099	0.076	-0.087	-0.016	0.626	0.287
KURTOSIS(G2)	1.466	-0.144	-0.295	-1.065	-0.868	-0.904	-0.757	-1.21	-0.925
SUM	863.37	912.45	1013.8	632.22	749.83	799.6	776.99	538.333	643.708
C.V.	0.183	0.216	0.306	0.584	0.393	0.378	0.331	1.113	0.576
MEDIAN	0.59	0.62	0.73	0.43	0.5	0.55	0.52	0.333	0.406

Table 3.4 :Statistics of Form 5 Trial Results

	BM F5T	BI F5T	MAT F5T	AMAT F5T	PHY F5T	CHEM F5T	BIO F5T	GRD F5T	AGG F5T	OFFER
NO OF CASES	1488	1488	1488	1488	1488	1488	1488	1488	1488	1488
MINIMUM	0	0.11	0	0	0	0	0	0	0	0
MAXIMUM	0.86	0.97	1	1	0.99	0.96	0.97	1	1	1
RANGE	0.86	0.86	1	1	0.99	0.96	0.97	1	1	1
MEAN	0.595	0.648	0.754	0.43	0.514	0.537	0.521	0.298	0.399	0.696
VARIANCE	0.012	0.017	0.039	0.062	0.037	0.041	0.026	0.128	0.053	0.212
STANDARD DEV	0.111	0.129	0.198	0.249	0.193	0.201	0.162	0.358	0.229	0.46
STD. ERROR	0.003	0.003	0.005	0.006	0.005	0.005	0.004	0.009	0.006	0.012
SKEWNESS(G1)	-0.802	-0.605	-1.157	0.101	0.022	-0.192	0	0.926	0.379	-0.853
KURTOSIS(G2)	1.495	0.335	0.753	-1.018	-0.822	-0.834	-0.429	-0.486	-0.605	-1.272
SUM	885.83	964.9	1122.34	639.83	765.27	798.58	775.91	444	593.688	1036
C.V.	0.186	0.2	0.263	0.58	0.375	0.375	0.31	1.2	0.575	0.661
MEDIAN	0.6	0.66	0.82	0.43	0.52	0.54	0.51	0.333	0.375	1

Note: BM = Bahasa Malaysia, BI = Bahasa Inggeris, MAT = Mathematics, AMAT = Additional Mathematics, PHY = Physics, CHEM = Chemistry, BIO = Biology, GRD = Grade, AGG = Aggregate, OFFER = Sixth Form Offer, F4 = Form 4 Final, F5M = Form 5 Mid-year, F5T = Form 5 Trial.

The mean and median values for all the variables were mostly about equal. This showed that the data had an acceptable central tendency as would be if the data were normally distributed. The histograms drawn for each subject showed that all are fairly normally distributed (Appendix C).

The skewness coefficient for Bahasa Malaysia, Mathematics and Grades were higher than the other subjects. The negative values for Bahasa Malaysia and Mathematics show the tendencies of more students getting higher marks than lower marks. This may be because students are now more aware of getting good grades for Bahasa Malaysia as a poor grade for it will cause a student to get a Grade 2 even though he/she scored well for the other subjects. However, the skewness statistic for all the other variables were less than 0.8, thus the distribution was not noticeably skewed (Bourque & Clark, 1992). Its formula essentially says to cube the standard scores, sum them, and then calculate the average. It is usually used to measure the lack of symmetry of the distribution.

Science students are generally better at mathematics compared to non-science stream students. However, the same student may not do well in Additional Mathematics, as it is known to be much more difficult. That is one of the reasons why many of them scored well in Mathematics but not for Additional Mathematics. The positive skewness for Grades showed that more students achieved Grade 1 and 2 compared to Grade 3 and 4.

In order to test whether there were any outliers in the data, the kurtosis coefficient was calculated. Kurtosis is based on the fourth powers of the deviations from the mean. If there were outliers in the data, the kurtosis would be very large (Berk, 1994). From the table above, the kurtosis for all variables was reasonably small thus confirming the non-existence of too many outliers.

Correlation tests were also performed to study their correlation with the targeted result, i.e. the Sixth Form Offer. This test was done to measure the strength of the linear relationship between two variables. The results, shown in table 3.5 and 3.6, were as follows:

Table 3.5: The Spearman Correlation Coefficients

	BMF4	BIF4	MATF4	AMATF4	PHYF4	CHEMF4	BIOF4	GRDF4	AGGF4
OFFER	0.401	0.433	0.54	0.541	0.605	0.636	0.588	-0.693	-0.692
	BMF5M	BIF5M	MATF5M	AMATF5M	PHYF5M	CHEMF5M	BIOF5M	GRDF5M	AGGF5M
OFFER	0.478	0.527	0.573	0.588	0.622	0.631	0.65	-0.722	-0.701
	BMF5T	BIF5T	MATF5T	AMATF5T	PHYF5T	CHEMF5T	BIOF5T	GRDF5T	AGGF5T
OFFER	0.451	0.548	0.573	0.594	0.608	0.64	0.63	-0.716	-0.698

Table 3.6: The Pearson Correlation Coefficients

	BMF4	BIF4	MATF4	AMATF4	PHYF4	CHEMF4	BIOF4	GRDF4	AGGF4
OFFER	0.413	0.439	0.569	0.52	0.586	0.63	0.573	-0.718	-0.704
	BMF5M	BIF5M	MATF5M	AMATF5M	PHYF5M	CHEMF5M	BIOF5M	GRDF5M	AGGF5M
OFFER	0.503	0.535	0.609	0.581	0.614	0.636	0.644	-0.747	-0.719
	BMF5T	BIF5T	MATF5T	AMATF5T	PHYF5T	CHEMF5T	BIOF5T	GRDF5T	AGGF5T
OFFER	0.484	0.565	0.6	0.587	0.605	0.65	0.62	-0.749	-0.717

Both the Spearman's and Pearson's Correlation Coefficient for all subjects was quite high except for B.M.. This showed a high and satisfactory correlation between these subjects towards predicting the targeted result, which in this case was the offer to continue studying in Form 6 Science. Omitting some of these subjects were definitely results in a lower accuracy of prediction. All the mathematics and pure science subjects were shown to be highly correlated. This necessitated the inclusion of these subjects as the aim of this project was to predict science stream students who would most likely do well in their SPM in order to be offered a place in Form 6 Science early.

The low correlation coefficient for B.M. could possibly be due to the extra emphasis placed on this important subject. Students who did well in other subjects but did poorly in B.M. will not be offered good grades. Students who did poorly in other subjects but performed well in B.M., on the other hand, may be offered satisfactory grades. This is the most probable reason for its low correlation coefficient. However, this subject has to be used, as it is a strong determinant whether a student achieves Grade 1 or not.

3.3 Research Procedures

3.3.1 Application of Approvals to Proceed with the Project

It was important that proper procedures were followed in the execution of this project to prevent any unnecessary problem later on. Before the researcher could proceed with this project, a letter of approval had to be secured from the Educational Planning & Research Division, Ministry of Education (EPRD). This letter was needed by the researcher to ask for permission from the State Director of Education to visit schools to gather data samples for this project. A visit to the department located in Pusat Bandar Damansara, Kuala Lumpur had to be done first to inform the relevant officer of the researcher's intention. Two sets of special forms were given to be filled up. On submitting these forms, the letter of approval was sent to the researcher after a week of waiting (Appendix D).

Next, a visit to the State Education Department was taken to seek the State Director of Education's permission. Another letter was then given to the researcher after a few days (Appendix E). Armed with these letters, the researcher then made the round at the targeted secondary schools situated around the Kinta district.

3.3.2 Preprocessing of Data

At the pre-processing stage of the data, all subjects' marks were scaled by dividing by 100. This is because the minimum mark is 0 and the maximum mark is 100. Each mark was graded from 1 to 9 based on the following criteria as shown in Table 3.7.

Table 3.7: Grading Criteria

Marks	Grades
80-100	1
70-79	2
65-69	3
60-64	4
55-59	5
50-54	6
45-49	7
40-44	8
0-39	9

The aggregate was calculated by adding the grades of the 6 best subjects. This resulted in a value that ranges from a minimum of 6 to a maximum of 54. In order to scale this, formulae (3.1) was used:

$$\frac{x - x_{\min}}{x_{\max} - x_{\min}}$$

where $x_{min}= 6$ and $x_{max}= 54$. (3.1)

The examination Grade was awarded based on a complex set of criteria. A candidate must secure a credit in Bahasa Malaysia first and a stated number of credits from the other subjects to be awarded a Grade 1. A candidate could only be awarded a Grade 2 even if he did well in the other subjects but secured only a pass in Bahasa Malaysia. A Grade 4 or SAP will be awarded if he failed his Bahasa Malaysia. The Grade that a candidate will be awarded is also dependent on the aggregates scored. For the purpose of simplifying the processing of data for this project, the researcher grouped all those candidates getting Grade 4, SAP and failed in the examination into Grade 4.

In order to scale this, formulae (3.2) was used:

$$\frac{x - x_{min}}{x_{max} - x_{min}}$$

$$\text{where } x_{min} = 1 \text{ and } x_{max} = 4. \qquad (3.2)$$

However, some students did not take all the subjects chosen for this project. Most who did not take the whole combination usually dropped either Physics or Additional Mathematics and substituted them with other subjects like Economics or Principals of Accounts. A much smaller number of students dropped Biology. Thus, the data samples of these students were not complete. However, using an option available in the Neuron Connection software, these empty fields were substituted with the mean values of the whole population.

CHAPTER FOUR

TRAINING RESULTS AND DISCUSSION

4.0 Overview of Chapter

This chapter discussed in detail the steps taken during the training of the neural networks where the best number of Hidden Units, the best Learning Coefficient, the best rate of Momentum, the best Activation Function, the best Training Algorithm, and the best Weights Distribution were identified. Training results were recorded in tables, bar charts and line graphs to offer the reader a clearer picture. Further testing were also performed using the trained model to determine the significance of other factors, like gender and types of schools, that might affect the accuracy of the networks.

A brief explanation on the usage of the trained neural networks model for prediction was also discussed at the end of the chapter. This part will help the reader to understand the practical use of the trained neural networks of this project.

4.1 Neural Networks Training and Testing Results

The Modeling and Forecasting tool used for this project is the Multi-Layer Perceptron Model. This neural network tool is optimized for prediction applications. The training process was done by performing a series of iterations to search for the best number of Hidden Units, the best Learning Coefficient, the best rate of Momentum, the best Activation Function, the best Training Algorithm, and the best Weights Distribution. The strategy used was to perform the first series of training by selecting a suitable number of hidden units together with the other starting values for the other parameters as offered by the Neuron Connection Software Package.

By changing only the values of the targeted parameter, which in this case was the number of hidden units, and repeating the training, the best two results were selected for further training. By varying the Seed number, further training were performed for these two selected values. The average for the percentage accuracy were counted and compared. The one that gave the higher of the two was then chosen. This were then repeated for the other parameters.

The starting parameters were as follows (Table 4.1):

Table 4.1: Starting Parameters before starting training of Neural Networks

Number of Hidden Units	8
Learning Coefficient	0.9
Rate of Momentum	0.1
Activation Function	Sigmoid
Learning Algorithm	Steepest Descent
Weights Distribution	Uniform
Seed Number	1

Firstly, the training was done to select the best number of Hidden Units.

4.1.1 Identifying the number of Hidden Units (HU)

Training was done by varying the number of hidden units starting from 8 until 14. Table 4.2 showed the results generated from the training exercise.

Table 4.2: Comparing percentage accuracies using different numbers of Hidden Units

No. of HU	Training	Testing
8	91.09	86.58
9	91.68	87.92
10	**91.93**	**87.25**
11	**91.85**	**87.25**
12	91.26	86.58
13	91.26	86.58
14	91.26	87.25

The bar chart (Figure 4.1) was plotted using Percentage Accuracy against Training and Testing Samples for HU = 8 to 14.

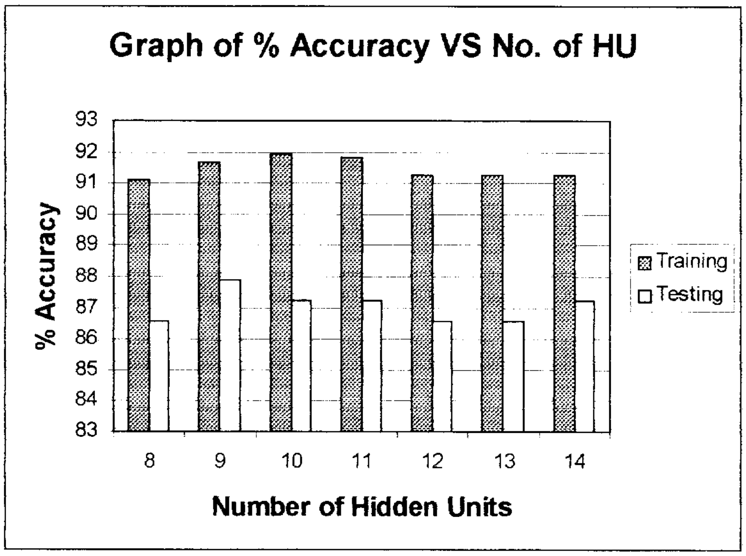


Figure 4.1: Percentage Accuracies for the various numbers of Hidden Units

The results shown in the table and graph above suggested that HU= 10 and HU = 11 gave higher percentage accuracy than the rest. These two best results were chosen for further testing to select the better of the two. This was done by repeating the training exercise by changing the seed number. Table 4.3 showed the results.

Table 4.3: Identifying the better of two numbers of Hidden Units

Seed	HU = 10		HU = 11	
No.	Training	Testing	Training	Testing
1	91.93	87.25	91.85	87.25
2	91.43	85.23	91.34	86.58
3	91.26	86.58	91.43	85.23
4	91.60	86.58	91.26	85.91
5	91.76	86.58	91.51	87.25
6	91.60	85.91	91.18	87.25
7	91.93	85.23	91.26	86.58
8	91.43	87.25	91.43	86.58
9	91.43	87.25	91.26	87.25
Average	**91.60**	**86.43**	91.39	86.25

A graph (Figure 4.2) was plotted to give a better graphical view of the results generated. The comprehensive testing by varying the seed number from 1 to 9 is shown in the table and graph above. As the average for HU = 10 was higher than the average for HU = 11, HU = 10 was chosen as the number of hidden units that will offer the best percentage accuracy.

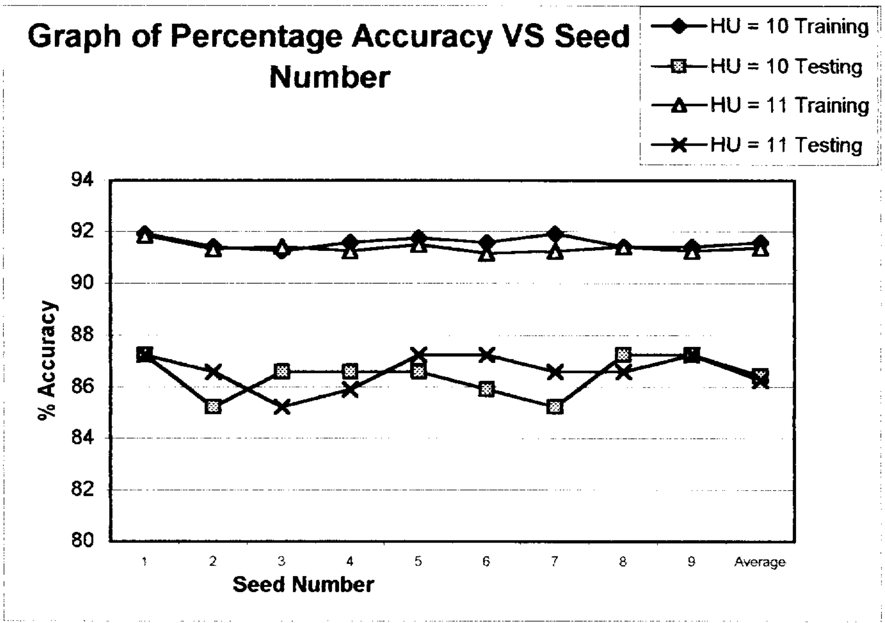


Figure 4.2: Percentage Accuracy vs Seed Number for different numbers of Hidden Units

Next, training was done to select the best Learning Coefficient.

4.1.2 Identifying the Learning Coefficient (LC).

By fixing the number of hidden units to 10, further training was done by varying the learning coefficient from 0.1 to 0.9. Table 4.4 showed the results achieved.

Table 4.4: Comparing percentage accuracies using different Learning Coefficients

LC	Training	Testing
0.1	91.43	85.23
0.2	91.43	85.91
0.3	91.18	86.58
0.4	91.18	86.58
0.5	91.68	85.91
0.6	91.68	85.91
0.7	91.60	86.58
0.8	**91.93**	**86.58**
0.9	**91.93**	**87.25**

The bar chart in Figure 4.3 was plotted using Percentage Accuracy against Training and Testing Samples for LC = 0.1 to 0.9.

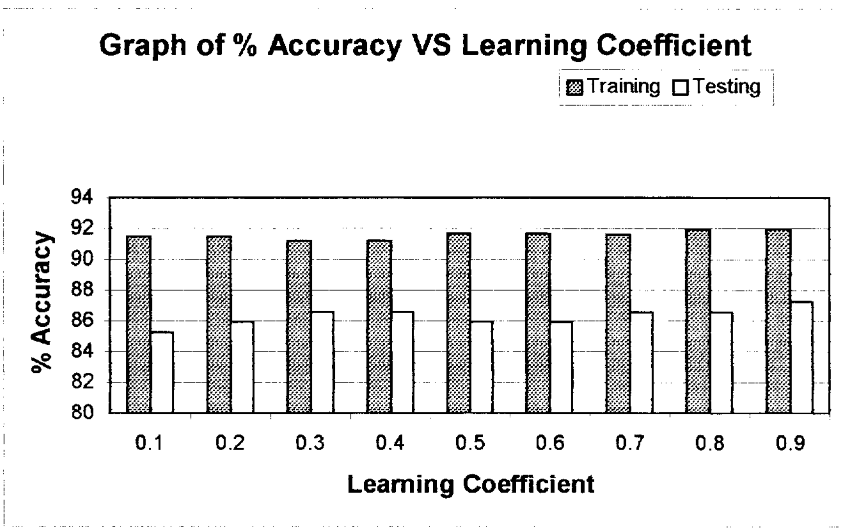


Figure 4.3: Percentage Accuracies for the various Learning Coefficients

The highest two results were achieved using LC = 0.8 and LC = 0.9. These two best results were selected for further testing to select the better of the two. Table 4.5 showed the results of the training exercise.

Table 4.5: Identifying the better of two Learning Coefficients

Seed	LC = 0.8		LC = 0.9	
No.	Training	Testing	Training	Testing
1	91.93	86.58	91.85	87.25
2	91.34	87.25	91.34	86.58
3	91.43	87.25	91.43	85.23
4	91.26	87.25	91.26	85.91
5	91.43	86.58	91.51	87.25
6	91.60	87.25	91.18	87.25
7	91.60	87.25	91.26	86.58
8	91.26	87.25	91.43	86.58
9	91.34	85.91	91.26	87.25
Average	91.47	86.95	**91.39**	**86.25**

A graph (Figure 4.4) was plotted to offer a better graphical view to allow for better comparison.

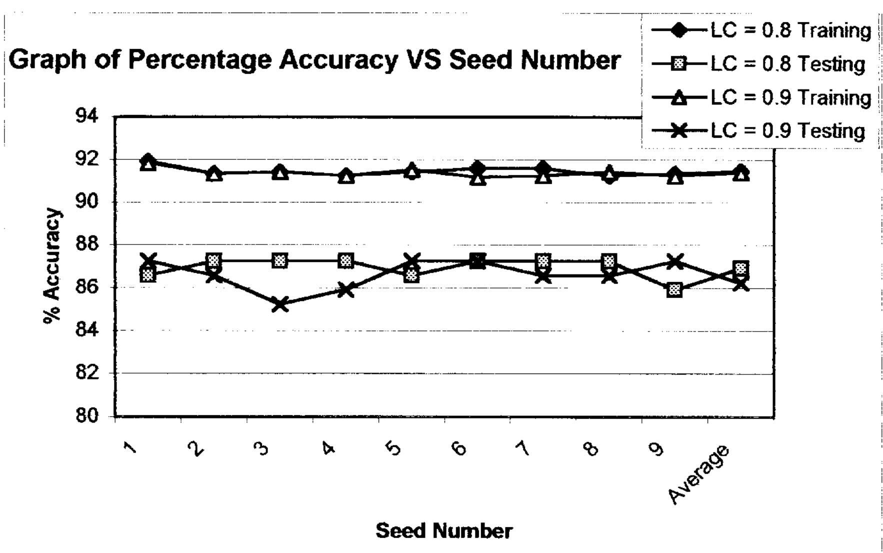


Figure 4.4: Percentage Accuracy vs Seed Number for the different Learning Coefficients

The comprehensive testing by varying the seed number from 1 to 9 is shown in the above table results. As the average for LC = 0.9 was higher than the average for LC = 0.8, LC = 0.9 was chosen as the Learning Coefficient that will offer the best percentage accuracy.

Next, training was done to select the best rate of Momentum.

4.1.3 **Identifying the Rate of Momentum (Mom).**

By fixing the number of hidden units to 10 and the learning coefficient to 0.9, further training was done by varying the rate of momentum from 0.1 to 0.9. Table 4.6 showed the results.

Table 4.6: Comparing percentage accuracies using different rates of Momentum

Mom	Training	Testing
0.1	**91.93**	**87.25**
0.2	91.85	87.25
0.3	**92.02**	**86.58**
0.4	91.26	87.25
0.5	91.34	86.58
0.6	91.51	85.23
0.7	91.26	87.25
0.8	91.26	87.25
0.9	91.18	87.25

The bar chart in Figure 4.5 was plotted using Percentage Accuracy against Training and Testing Samples for rate of Momentum = 0.1 to 0.9.

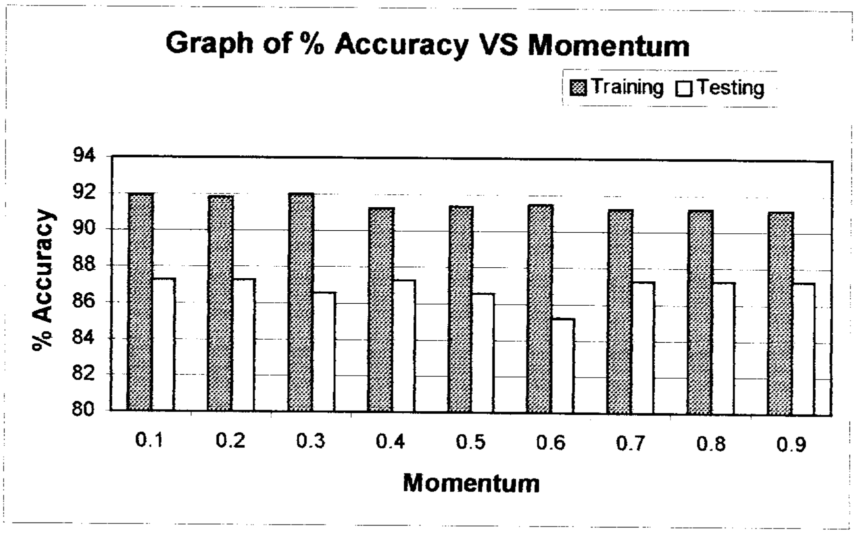


Figure 4.5: Percentage Accuracies for the various rates of Momentum

The best results were achieved using Mom = 0.1 and Mom = 0.3. These two best results were selected for further testing to select the better of the two. Table 4.7 showed the results.

Table 4.7: Identifying the better of two rates of Momentum

Seed No.	Mom = 0.1		Mom = 0.3	
	Training	Testing	Training	Testing
1	91.85	87.25	92.02	86.58
2	91.34	86.58	91.51	85.23
3	91.43	85.23	92.02	87.92
4	91.26	85.91	91.68	87.25
5	91.51	87.25	92.02	86.58
6	91.18	87.25	91.43	87.92
7	91.26	86.58	91.68	87.25
8	91.43	86.58	92.18	86.58
9	91.26	87.25	91.43	87.25
Average	91.39	86.25	**91.77**	**86.95**

A graph (Figure 4.6) was plotted to offer a better graphical view to allow for better comparison. The comprehensive testing by varying the seed number from 1 to 9 is shown in the above table results. As the average for Mom = 0.3 was higher than the average for Mom = 0.1, Mom = 0.3 was chosen as the Momentum that will offer the best percentage accuracy.

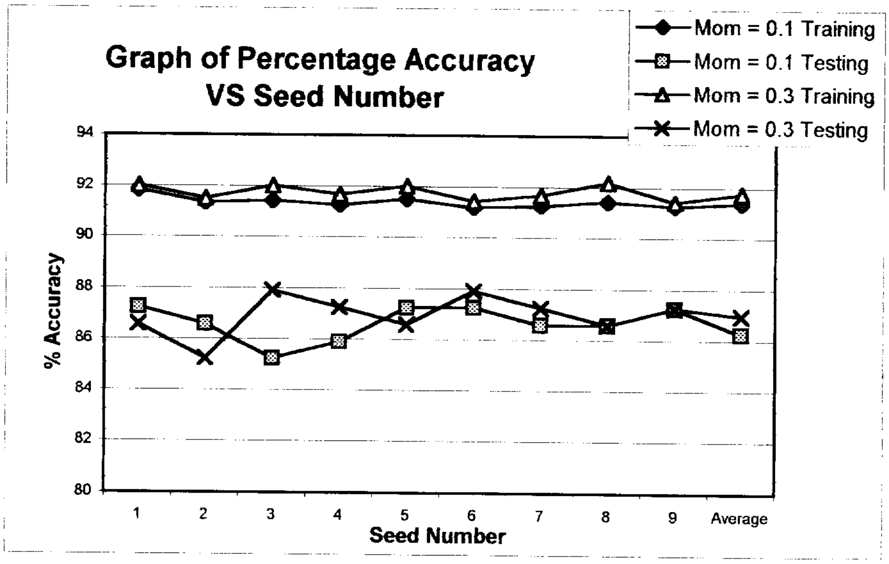


Figure 4.6: Percentage Accuracy vs Seed Number for the different rates of Momentum

Next, training was done to select the best Activation Function. This was done by repeating the training exercise after changing the three different activation functions available.

4.1.4 Identifying the Activation Function.

By fixing the number of hidden units to 10, the learning coefficient to 0.9 and the rate of momentum to 0.3, further training was done by varying the activation function. Table 4.8 showed the result.

Table 4.8: Comparing percentage accuracies using different Activation Functions

Activation Function	Training	Testing
Sigmoid	**92.18**	**86.58**
Linear	91.34	86.58
Tanh	91.60	87.25

The bar chart in Figure 4.7 was plotted using Percentage Accuracy against Training and Testing Samples for the three different types of Activation Functions.

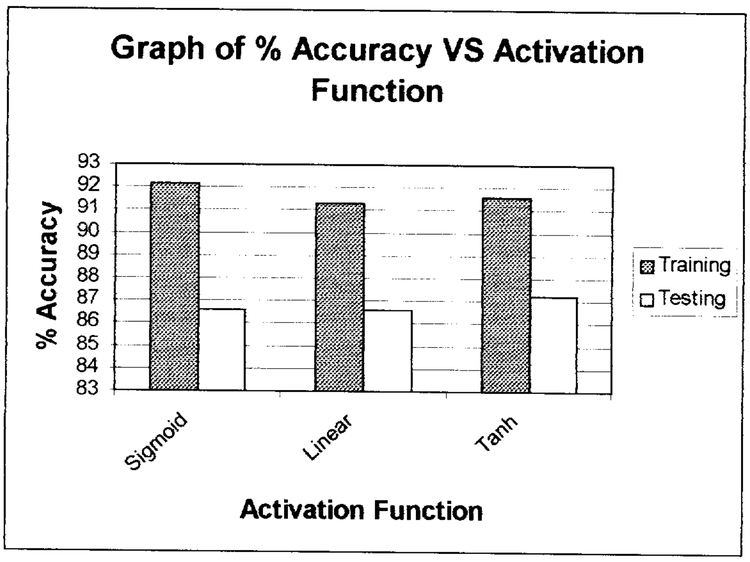


Figure 4.7: Percentage Accuracies for the various Activation Functions

The bar chart showed that an overall higher percentage was achieved by using the sigmoid function. Therefore, Sigmoid function was selected as the best.

The next step was to select the best learning algorithm.

4.1.5 **Identifying the Learning Algorithm.**

By fixing the number of hidden units to 10, the learning coefficient to 0.9, the rate of momentum to 0.3 and the sigmoid activation function, further training was done by varying the learning algorithm. Table 4.9 showed the result.

Table 4.9: Comparing percentage accuracies using different Learning Algorithms

Learning Algorithm	Training	Testing
Steepest Descent	**92.18**	**86.58**
Conjugate Gradient	91.18	87.25

The result showed that a better percentage accuracy was achieved using Steepest Descent. This resulted in selecting Steepest Descent as the best learning algorithm.

Lastly, training was performed to select the best weights distribution method.

4.1.6 **Identifying the Weights Distribution.**

By fixing the number of hidden units to 10, the learning coefficient to 0.9, the rate of momentum to 0.3, the Sigmoid activation function and the Steepest Descent learning algorithm, further training was done by varying the Weights Distribution method. Table 4.10 showed the result.

Table 4.10: Comparing percentage accuracies using different Weights Distributions

Weights Distribution	Training	Testing
Uniform	**92.18**	**86.58**
Gaussian	91.51	85.91

The result showed that a better percentage accuracy was achieved using Uniform Weights Distribution. Therefore, the Uniform weights distribution method was selected as the best.

4.2 Comparing Results of Other Factors Using Chosen Neural Networks Model

From the results achieved in all the training iterations performed as mentioned above, the following summarizes the best combination of parameters for the best NN model.

The best NN model was thus achieved using the following parameters as shown in Table 4.11.

Table 4.11: The chosen Parameters for this project's Neural Networks Model

Best number of Hidden Units	10
Best Learning Coefficient	0.9
Best Rate of Momentum	0.3
Best Activation Function	Sigmoid
Best Learning Algorithm	Steepest Descent
Best Weights Distribution	Uniform
Best Seed Number	8

By using this combination, further training was performed on data samples of the 10 individual schools. Table 4.12 showed the different results achieved.

Table 4.12: Training results of different types of schools

Schools	Training	Testing	Type
A	92.08	100.00	Girls, Urban
B	88.89	95.45	Girls, Urban
C	88.37	85.71	Girls, Urban
D	87.30	87.50	Girls, Rural
E	89.58	83.89	Girls, Urban
F	93.35	88.31	Girls, Urban
G	81.19	84.62	Boys, Rural
H	95.52	96.43	Boys, Urban
I	95.71	84.09	Boys, Urban
J	96.86	90.28	Boys, Urban
Overall	92.18	86.58	Combined

The training and testing results of most individual schools were either comparatively better or on par with the overall data samples. This slight difference in results was to be expected as different schools have slightly different characteristics with respect to different grading methods, different level of difficulties in their examinations papers and different ways of marking answers.

The percentage accuracy achieved for both urban and rural type schools were both quite close to the overall results and also to each other. However, the results from the urban schools were slightly better. This slight difference, however, is too small to be able to cause a major drop or rise in prediction accuracy. This showed a lack of determinant as to the significant of data samples from urban and rural schools affecting the NN model. Thus, it can now be concluded that there is no significant different between data samples from urban or rural schools. The same NN model is suitable to be applied to both types of schools.

Comparison of results from girls' schools with boys' schools showed that they were not different from each other. However, the results from the boys' schools were marginally better. Again, this slight difference is not large enough to make a concrete conclusion as to the significant of data samples from boys' and girls' schools affecting the NN model. Again, the conclusion is that data samples from both boys' and girls' schools will not severely affect the outcome of the NN model.

Further training and testing were performed by using only data samples from different Forms and also from a combination of two Forms. The results were as follows (Table 4.13):

Table 4.13: Training results for different sets of examinations

Form	Training	Testing
Form 4 only	81.85	81.48
Form 5 Midyear only	82.00	80.25
Form 5 Trial only	82.62	79.01
Form 5 Midyear & Trial only	86.31	72.84
Form 4 & Form 5 Trial only	86.00	80.25
Form 4 & Form 5 Midyear only	84.62	85.19
Overall	92.18	86.58

The results showed that a much lower prediction accuracy was achieved when data samples from only a single Form was used. The percentages increased slightly when data samples were used from the combination of any two Forms. However, these results were significantly much lower when compared with the overall result when all three sets of examination results from Form 4 and Form 5 were used. This showed a rather significant determinant of Form 4 and Form 5 examination results in affecting the eventual accuracy of the NN model. As such, it was concluded that the inclusion of all subjects marks from the Form 4, Form 5 Mid-year and the Form 5 Trial examinations were included in the final data samples.

4.3 Neural Networks Model for Prediction

In order to use the trained NN model for prediction, a new data set containing the new candidates' marks, aggregates and grades similar in format to the original development data set must be created first. However, the last field for Sixth Form offers will be empty. This is because these new candidates, as they should have just completed their examinations, will not have any information as to whether they have an offer to continue Sixth Form or not. The purpose of this NN model is to come out with a very high accuracy of prediction as to whether these candidates should be offer a place in Sixth Form or not before the SPM results is out.

When using Neuron Connection, the original input development data set needed to be saved in <filename>.nna, <filename>.csv, <filename>.txt, <filename>.sav, <filename>.sys, or <filename>.xls formats. For this project, the researcher used the <filename>.csv format as it was easily done using Microsoft's Excel. Microsoft's Excel has an option that allowed you to save your file as <filename>.csv. The user had to save the best NN model after training was completed. This best NN model was saved as a <filename>.nni file that could be opened again whenever needed.

This new input run data set must be saved in the same format too. However, Neuron Connection does not allow the last field to be empty, as should be because these candidates would not have this information as yet, and should be filled in with some dummy values (the researcher used 1s'). You could then load this new run data set from the data input tool as shown in Figure 4.8.

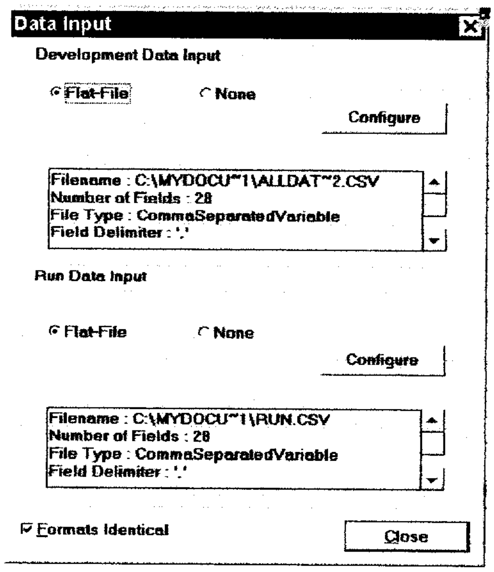
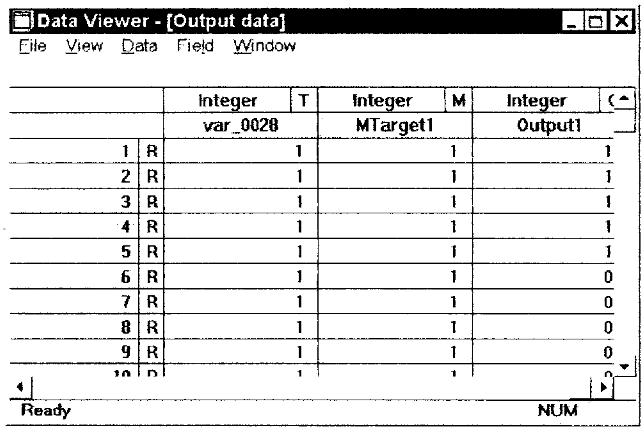


Figure 4.8: To load run data input from the data input tool

From the data output tool, you could then specified the name and type of file for the prediction to be written. In order to run the prediction application, the user would have to view the data set from the data output tool and choose Run from the view menu. A sample of the prediction's results is shown in Figure 4.9. The Output column showed the result of the prediction.



		Integer	Integer	Integer
		var_0028	MTarget1	Output1
1	R	1	1	1
2	R	1	1	1
3	R	1	1	1
4	R	1	1	1
5	R	1	1	1
6	R	1	1	0
7	R	1	1	0
8	R	1	1	0
9	R	1	1	0

Figure 4.9: Prediction results using data output tool

This prediction could be performed by using a school data as a single data set or by dividing the data set into smaller sets. It is suggested that prediction could be done on a class by class basis as this would be easier to manage and faster.

CHAPTER 5
CONCLUSIONS AND RECOMMENDED FURTHER STUDY

5.1 Conclusions

The high success rate results achieved, in the order of 90+% accuracy, in correct prediction of students to be admitted early to Sixth Form illustrated the ability of neural networks using MLP to perform the task of selecting correct students for early intake. The findings of this project as to the suitability of the trained model to be used for most schools, irrespective of genders and whether urban or rural, promised an encouraging use of neural networks in the near future. This positive result, hopefully, will encourage other similar researches to study the use of neural network in the solving of other similar students' early intake problems.

The success of this project, in term of getting a high percentage of accuracy in prediction, depended very much on the availability of complete and quality data from the various targeted schools. It is hoped that an up-to-date and complete record of students' academic achievements will be stored and be accessed readily when needed to ensure the smooth running of further similar projects. Schools are advised to start keeping their school based academic results stored in computer based files in line with the modernization of the education system.

5.2 Recommended Further Study

It is the hope of the researcher to offer this solution as an answer to the on-going problem affecting students admission to Sixth Form. It is also hoped that this project can be extended to include students from the non-science stream too to present a comprehensive solution to this problem. That, however, will need a full-scale project that will need much more further research and study.

The capability of neural networks in recognizing trends in raw data and using this identified trends for the purpose of prediction could thus be studied further to utilize and maximize its potential. More in line with the findings of this project, neural networks using MLP could also be trained and developed for other similar uses in the prediction of students' results for other types of early intakes. These include selecting successful students for early intake into advance level courses in private colleges which usually start in January each year, pre-university courses that also start before the SPM or STPM results are announced, and selecting and admitting students into the Matriculation classes based on their forecast results.

Enterprising companies could also implement similar neural networks to select potential employees to be trained further to fit into their business operations. This likely scenario is probably attractive to these companies, as they will be able to select first, the better of these many new employees that enter into the labour markets yearly. This is because they could select them first even before these new workers have their diplomas and degrees ready.

However, more could also be done to study the uses of neural networks with respect to educational data, such as this project. For example, the researcher made a trip to the Educational Planning and Research Division's (EPRD) library in September, 2000 in an attempt to research on past thesis of this type but found none.

As neural networks is more than just using MLP, using other modeling and forecasting tools widely available would most probably offer equal, if not better, results. In particular, modeling and forecasting tools like Radial Basis Function, Bayesian Network Tool, and Kohonen Network Tool that are all available in the Neuron Connection Software Package used in this project may offer much promises.

REFERENCES

Ainsworth, W.A. and Warren, N.P.. Applications of multilayer perceptrons in text-to-speech synthesis systems. In Neural networks for vision, speech and natural language. Edited by R. Linggard, D.J. Myers and C. Nightingale (1992). Chapman & Hall.

Bhat, N.V., Minderman, Jr. P.A., McAvoy, T., and Wang, N.S. (1990). Modelling chemical process systems via neural computation. In Neural networks current applications, edited by P.G.J. Lisboa (1992). Chapman & Hall.

Bourque, L.B., and Clark, V.A. (1992) Processing Data: The Survey Example. Sage University Paper series on Quantitative Applications in the Social Sciences, 07-085. Newbury Park, CA: Sage.

Berk, K.N. (1994) Data Analysis with Student SYSTAT. Cambridge, MA.: Course Technology, Inc.

Caren Marzban, E. DeWayne Mitchell and Gregory J. Stumpf, 19-- , A Neural Network for Tornado Diagnosis, National Severe Storms Laboratory, Norman, OK 73069, Cooperative Institute for Mesoscale and Meteorological Studies, and Department of Physics, University of Oklahoma, Norman, OK 73019.

DARPA Neural Network Study 1988, AFCEA International Press.

Eberhart, R.C. and Dobbins, R.W. 1990. *Neural Network PC Tools; A Practical Guide*. Academic Press. California.

Fadzilah Siraj, 1994. A Survey of Pruning Algorithms in Neural Networks.

Freeman, J., and D. Skapura. 1991. Neural Networks. Reading MA: Addison-Wesley.

Freeman, J., and D. Skapura. 1992. Neural Networks. Algorithms, Applications, and Programming Techniques: Addison-Wesley.

Gurney, K. 1996. An Introduction to Neural Networks, Online version. Psychology Department. University of Sheffield. UK. http://www.shef.ac.uk/psychology/gurney/notes/index.html.

Haykin, S. 1994. Neural Networks, A Comprehensive Foundation, Macmillan College Publishing Co. Inc..

Hong, S and Frank C. Lin. Medical Diagnosis by the Virtual Physician. Proceedings of the 12th IEEE Symposium on Computer-Based Medical Systems. 1998. Institute of Electronic and Electronics Engineers, Inc.

Koekkoek, E. 1996. Development of Neural Network Models to Predict Soil Water Retention, Dept. of Soil Science and Geology, Wageningen.

Le Cun, Y., Boser, B., Denker, J.S., Henderson, D., Howard, R.E., Hubbard, W. and Jackel, L.D.. Handwritten digit recognition with a back-propagation network. In Neural networks current applications, edited by P.G.J. Lisboa (1992). Chapman & Hall.

Mason, J.S. and Andrews, E.C.J. . Dissection of perceptron structures in speech and speaker recognition. In Neural networks for vision, speech and natural language. Edited by R. Linggard, D.J. Myers and C. Nightingale (1992). Chapman & Hall.

McClelland , J., D. Rumelhart, and the PDP Research Group. 1986. Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1: Foundations. Cambridge, MA: MIT Press.

McClelland, J., and D. Rumelhart. 1988. Explorations in Parallel Distributed Processing. Cambridge, MA: MIT Press.

McCulloch, W., and W. Pitts. 1943. A logical calculus of the ideas imminent in nervous activity. Bulletin of Mathematical Biophysics 5: 115-33.

Minsky, M., and S. Papert. 1969. Perceptrons. Cambridge, MA: MIT Press.

Neural Connection 2.0 User's Guide, SPSS Inc. and Recognition Systems Inc. 1997.

Nigrin, A. 1993, Neural Networks for Pattern Recognition, Cambridge, MA: The MIT Press.

Parker, D. B. 1985, "Learning-logic", M.I.T. Cen. Computational Res. Economics Management Sci., Cambridge, MA, TR-47.

Robert J. Schalkoff 1997, Artificial Neural Networks, McGraw-Hill International Editions.

Rosenblatt, F. 1958. The perceptron: A probabilistic model for information storage and organization in the brain. Psychology Review, 65: pp. 386-408.

Rumelhart, D. E., Hinton, D. E. and Williams, R. J. 1986, "Learning representation by backpropagating errors", Nature 323(9), pp. 533-536.

Rumelhart, D., G. Hinton, and R. Williams. 1988. Learning internal representations by error propagation. In Neurocomputing, edited by J. Anderson and E. Rosenfeld, 675-695. Cambridge, MA: MIT Press.

Salchenberger,L et al. Using Neural Networks to Aid the Diagnosis of Breast Implant Rupture. Computers Operations Research V.24, No5, pp435-444, 1997.

Smyth, S.G.. Segmental sub-word unit classification using a multilayer perceptron. In Neural networks for vision, speech and natural language. Edited by R. Linggard, D.J. Myers and C. Nightingale (1992). Chapman & Hall.

Tirri, H., Silander, T. and Tirri, K. (1997). Using Neural Networks for descriptive statistical analysia of educational data. Annual meeting of the American Educational Research Association (Chicago, IL, USA, March 1997).

Vincent, J.M., Myers, D.J. and Hutchinson, R.A.. Image feature location in multi-resolution images using a hierarchy of multilayer perceptrons. In Neural networks for vision, speech and natural language. Edited by R. Linggard, D.J. Myers and C. Nightingale (1992). Chapman & Hall.

Wasserman, P. 1989. Neural Network Computing. New York: Van Nostrand Reinhold.

Widrow, B. and Hoff, M. E. 1960, "Adaptive switching circuits", WESTCON Convention, Record Part IV, pp. 96-104.

Werbos, P. J. November 1974, "Beyond regression: new tools for prediction and analysis in the behavioural sciences", PhD. dissertation, Committee on Appl. Math., Harvard Univ., Cambridge, M. A..

Werbos, P. J. October 1990, "Backpropagation through time: What it does and how it to do it", Proceedings of IEEE, VOL. 78, No. 10.

Woodland, P.C.. Spoken alphabet recognition using multilayer perceptrons. In Neural networks for vision, speech and natural language. Edited by R. Linggard, D.J. Myers and C. Nightingale (1992). Chapman & Hall.

Zurada, J.M. 1992, Introduction to Artificial Neural Systems, Boston: PWS Publishing Company.

APPENDIX A

The Neuron Connection Software Package User Manual

Neuron Connection is a registered trademark of SPSS Inc. The Neuron Connection Version 2 Software Package used in this project is a software system that allows you to build complex applications for solving various problems using neural computing and other techniques. Neuron Connection provides a graphical user interface as a workspace, the program window where you can build problem-solving applications.

The workspace allows you to build, train, and run analysis applications. It has tools for inputting data, statistical analysis, problem modeling, and producing results. The diagram in figure A.1 is a sample of this workspace.

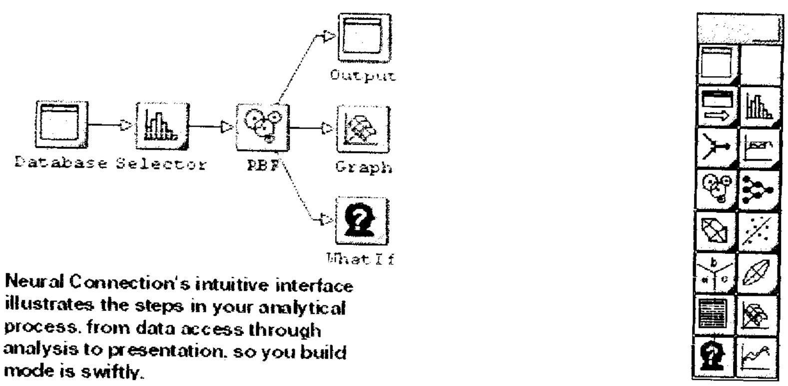


Figure A.1: Neuron Connection's Workspace

Tools are selected as icons from a palette (as shown on the right of diagram), and moved onto the workspace, where they can be connected to other tools. These connections determine an application's topology, and the path along which data flows.

The topology used in this project is as shown in the screen capture in figure A.2.

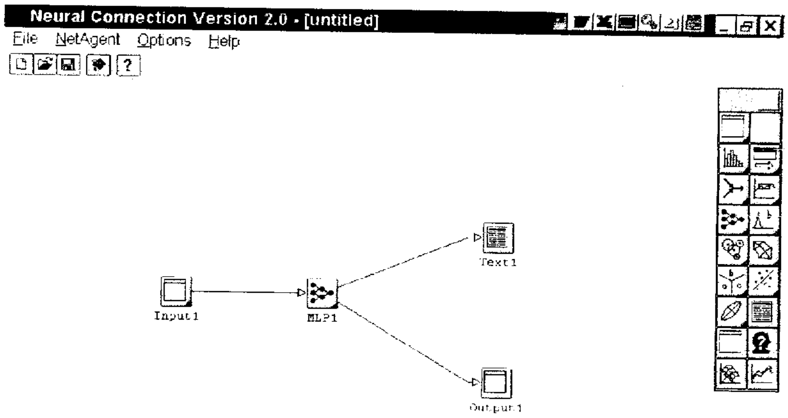


Figure A.2: Project's Topology

The data input tool can import data either from files or by cutting and pasting from other Windows applications. It can also be used to edit data. The data input tool's data viewer is shown in the figure A.3.

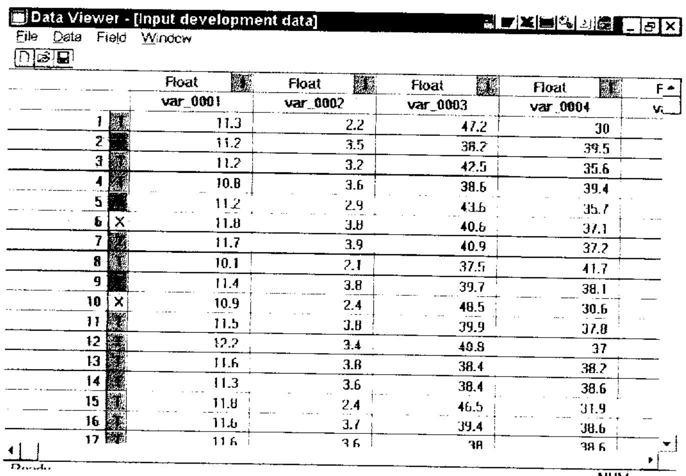


Figure A.3: Data Input Tool's Data Viewer

You can also select the data allocation for your data. The Data Allocation interface shown in figure A.4 is used to divide the data into training, validation and test data.

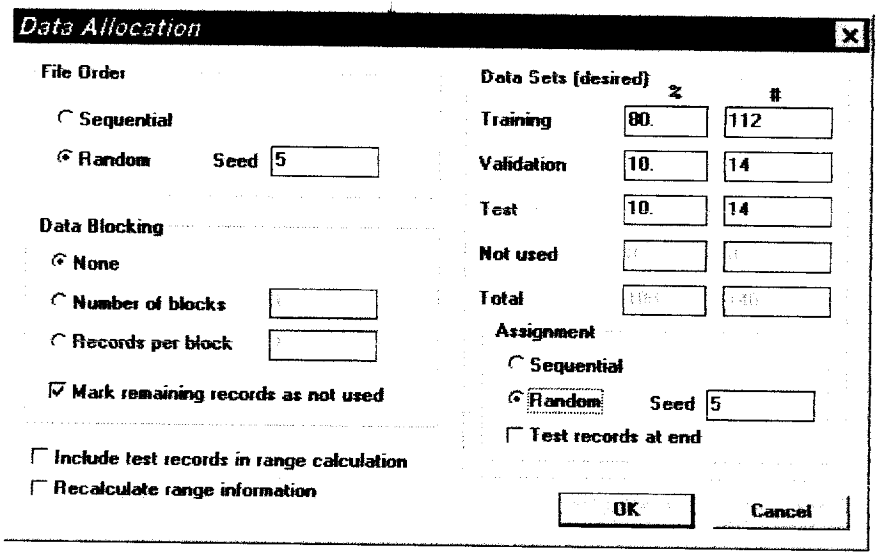


Figure A.4: Data Allocation Interface

In order to train and test the new application, the data input tool dialog box (Figure A.5) allows you to open the development data file. After the application has been trained, the run data file will be used to implement the trained model.

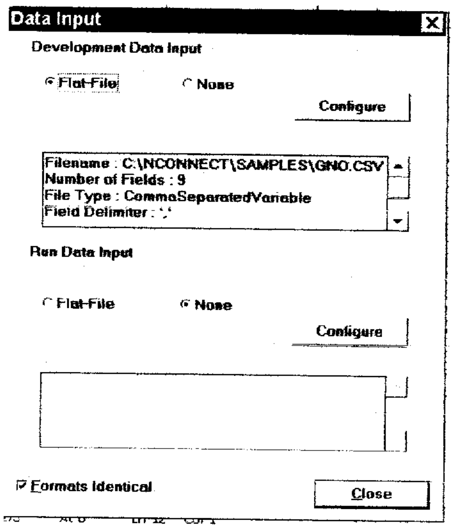


Figure A.5: Data Input Tool Dialog Box

The Multi-Layer Perceptron modeling and forecasting tool icon is shown below. This is the modeling tool that is optimized for prediction applications.

Some parameters (as was discussed in chapter 4) that will affect the performance of the MLP can be selected here. The MLP dialog box is shown in figure A.6.

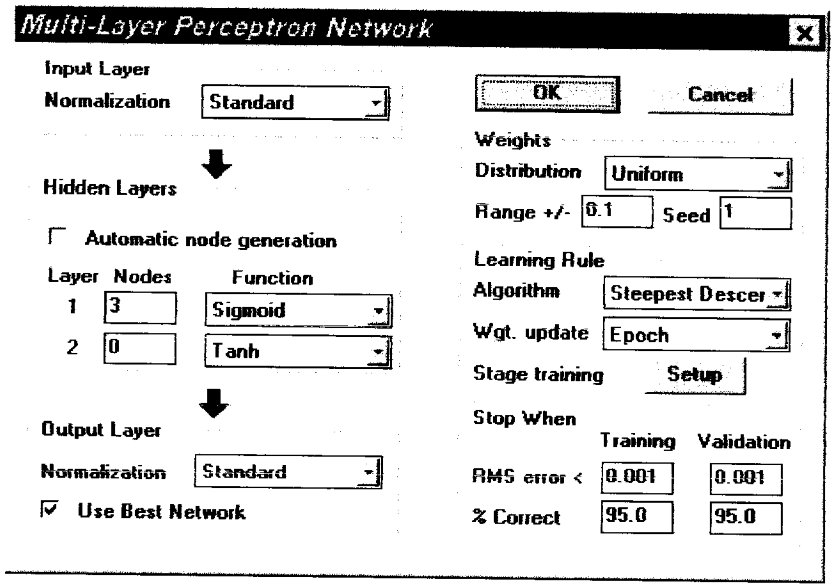


Figure A.6: MLP Dialog Box

While a training exercise is being performed, an error-time graph will be displayed as shown in figure A.7. You can monitor the progress of the training from here.

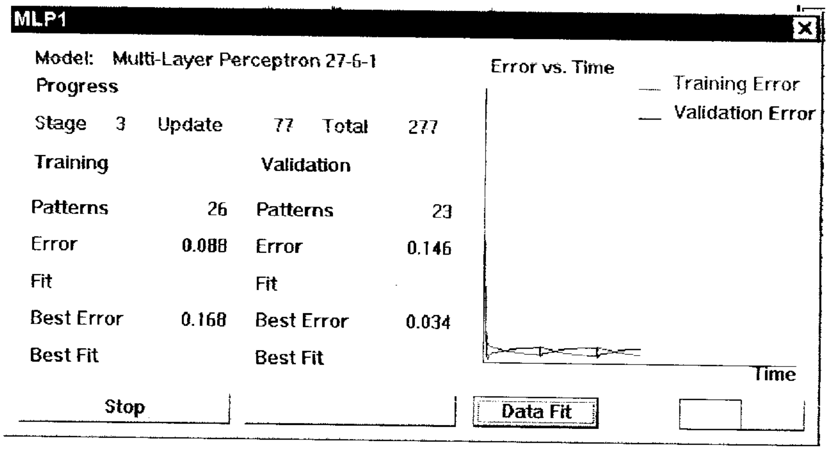


Figure A.7: Error-Time Graph

Training will be terminated when the desired percentage error is achieved or when the user stop it pre-maturely.

The data output tool can export data either to files, display training results as text, and show the success rate. The data output tool's data viewer and the cross tabulation table are shown in figure A.8 and figure A.9.

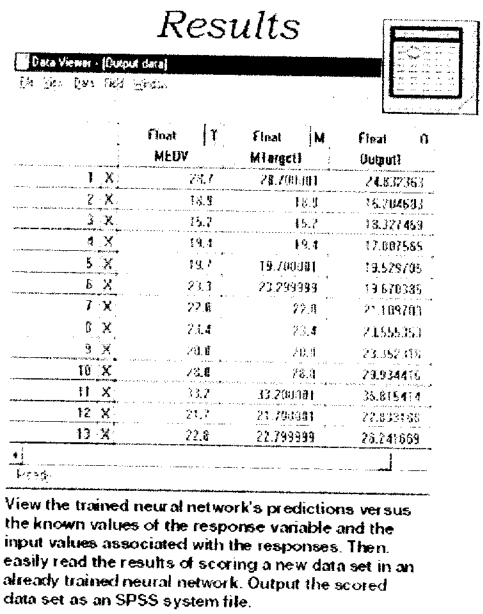


Figure A.8: The Data Output Tool's Data Viewer

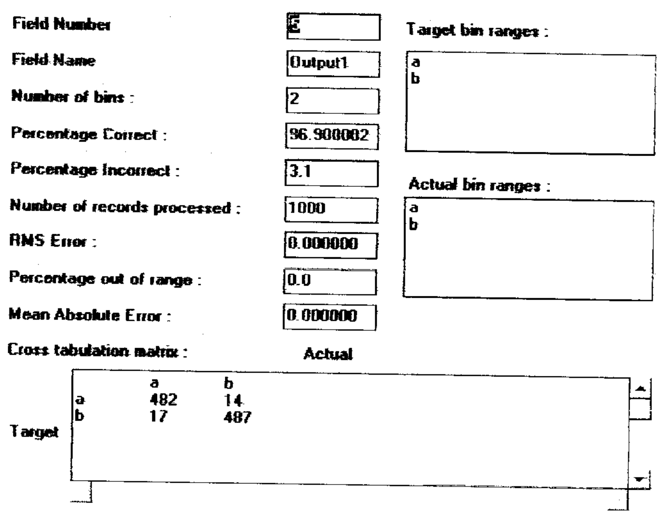


Figure A.9: The Data Output Tool's Cross Tabulation Table

The text output tool can also export data either to files, displays results as text, and shows the success rate. The text output tool's data viewer is shown in figure A.10.

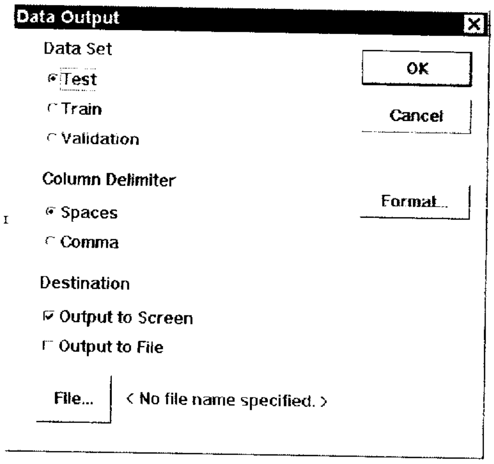


Figure A.10: The Text Output Tool's Data Viewer

The training results will be displayed in a cross tabulation matrix as shown in Figure A.11. The percentage correct figure shown here is the result of a training exercise.

```
** Cross Tabulation Matrix For Output 1 **

True                 Predicted
----                 ---------
                112.0+        174.666667+ 237.333333+
112.0+          0             0           0
174.666667+     0             5           2
237.333333+     0             4           29

Total number of targets : 40

Total correct : 34

Percentage correct : 85.00%
```

Figure A.11: The Text Output Tool's Cross Tabulation Matrix

APPENDIX B

The Data Input Form

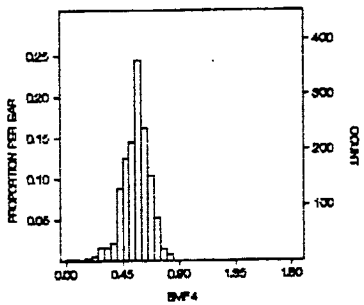
Form 4(Final Examinations)									Form 5 (Mid-Term Examinations)									Form 5 (Trial SPM)									Sixth Form Offer
B.M.	B.I.	Math	Amath	Phy	Chem	Bio	Grd	Agg	B.M.	B.I.	Math	Amath	Phy	Chem	Bio	Grd	Agg	B.M.	B.I.	Math	Amath	Phy	Chem	Bio	Grd	Agg	

APPENDIX C

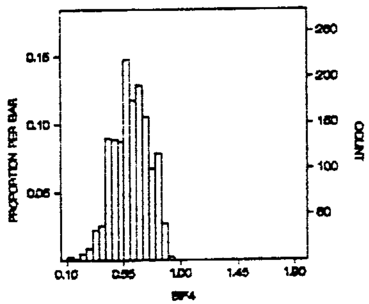
The Histograms for Form 4 and Form 5

Histogram

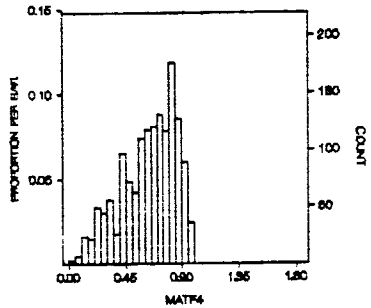
Histogram For Bahasa Malaysia Form 4 Final



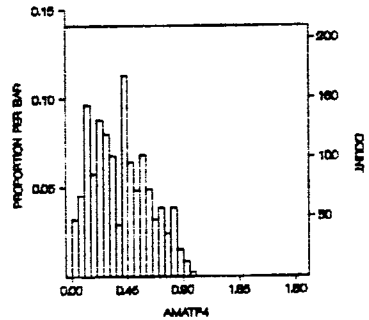
Histogram For Bahasa Inggeris Form 4 Final



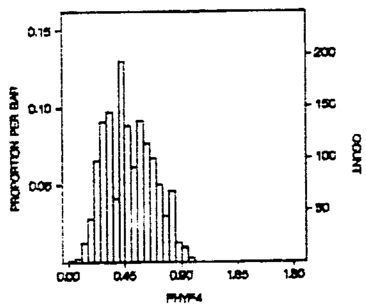
Histogram For Matematics Form 4 Final



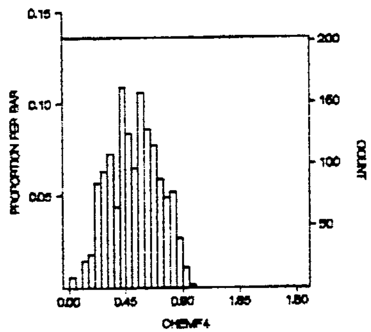
Histogram For Additional Mathematics Form 4 Final



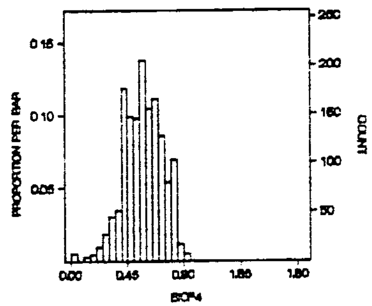
Histogram For Physics Form 4 Final



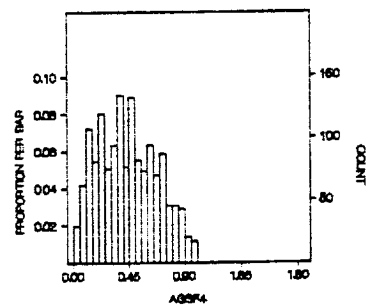
Histogram For Chemistry Form 4 Final



Histogram For Biology Form 4 Final

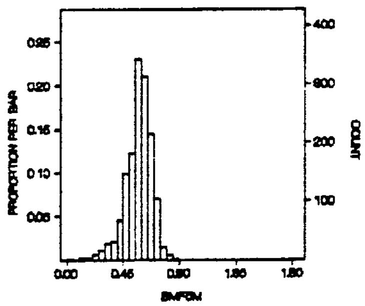


Histogram For Aggregates Form 4 Final

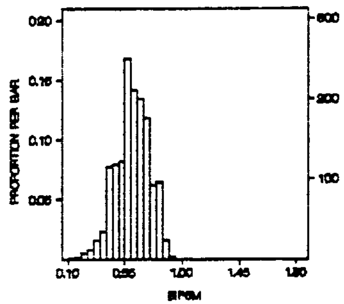


Histogram

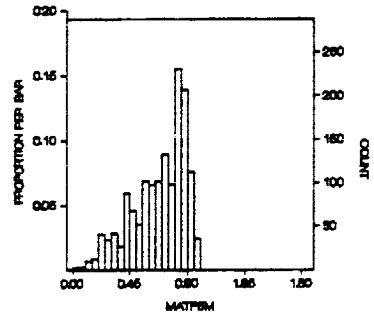
Histogram For Bahasa Malaysia Form 5 Mid-Year



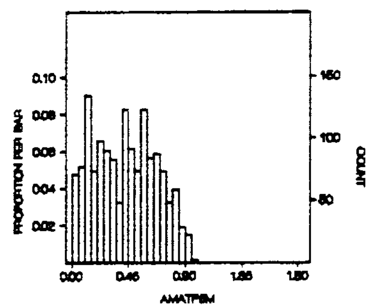
Histogram For Bahasa Inggeris Form 5 Mid-Year



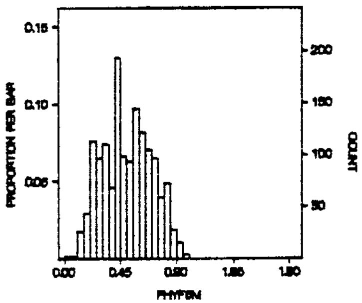
Histogram For Mathematics Form 5 Mid-Year



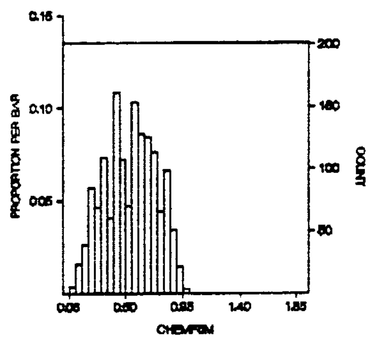
Histogram For Additional Mathematics Form 5 Mid-Year



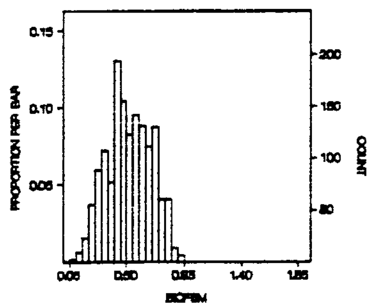
Histogram ForPhysics Form 5 Mid-Year



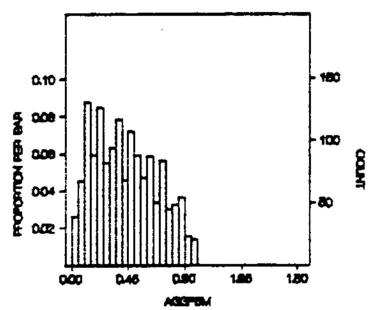
Histogram ForChemistry Form 5 Mid-Year



Histogram For Biology Form 5 Mid-Year

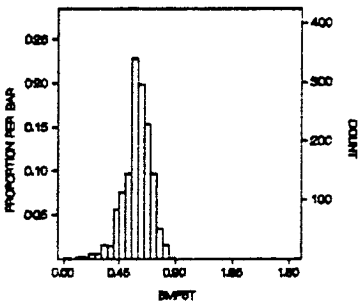


Histogram For Aggregates Form 5 Mid-Year

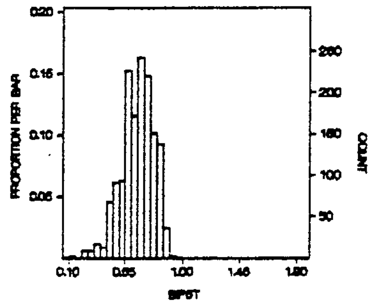


Histogram

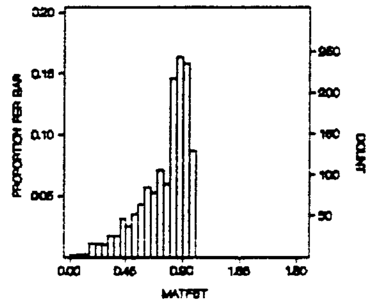
Histogram For Bahasa Malaysia Form 5 Trial



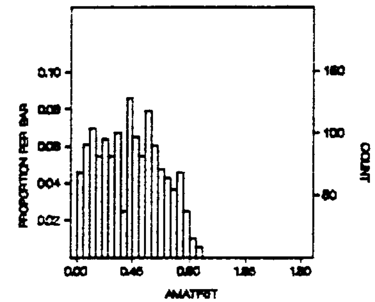
Histogram For Bahasa Inggeris Form 5 Trial



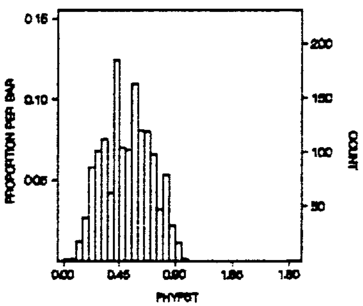
Histogram For Mathematics Form 5 Trial



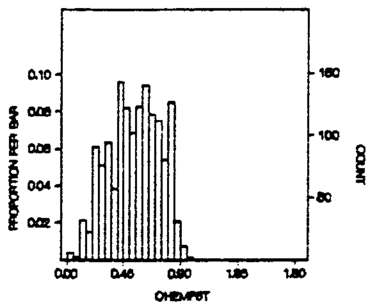
Histogram For Additional Mathematics Form 5 Trial



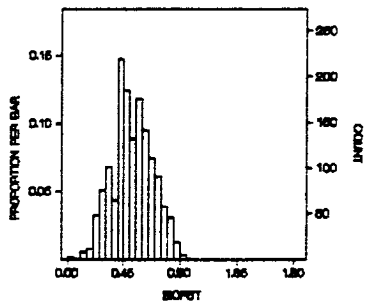
Histogram For Physics Form 5 Trial



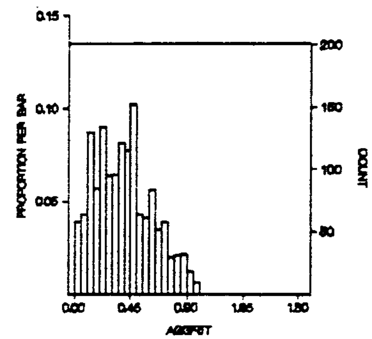
Histogram For Chemistry Form 5 Trial



Histogram For Biology Form 5 Trial



Histogram For Aggregates Form 5 Trial



APPENDIX D

The approval letter from The Educational Planning and Research Division (EPRD)

KEMENTERIAN PENDIDIKAN MALAYSIA
BAHAGIAN PERANCANGAN DAN
PENYELIDIKAN DASAR PENDIDIKAN
PARAS 2, 3 DAN 5, BLOK J
PUSAT BANDAR DAMANSARA
50604 KUALA LUMPUR

Telefon : 03 - 2586900
Fax : 03 - 2554960
Laman Web : http://eprd.kpm.my

Ruj. Kami : KP(BPPDP) 13/15 (302)
Tarikh : 31 Mei 2000.

En. Wong Tuck Sung,
83, Lengkok Canning,
Ipoh Garden,
31400 Ipoh,
PERAK.

Tuan,

Kebenaran Untuk Menjalankan Kajian Di Sekolah-Sekolah, Maktab-Maktab Perguruan, Jabatan-Jabatan Pendidikan Dan Bahagian-Bahagian Di Bawah Kementerian Pendidikan Malaysia

Adalah saya dengan hormatnya diarah memaklumkan bahawa permohonan tuan untuk menjalankan kajian bertajuk:

> **"The Selection of Successful Students for Early Admission Into Sixth Form Science Stream Using Neural Networks Model"**

telah diluluskan.

2. Kelulusan ini adalah berdasarkan apa yang terkandung di dalam cadangan penyelidikan yang tuan kemukakan ke Bahagian ini. **Kebenaran bagi menggunakan sampel kajian perlu diperolehi daripada Ketua Bahagian/Pengarah Pendidikan negeri yang berkenaan.** Sila kemukakan ke Bahagian ini senaskah laporan kajian tuan setelah ia selesai kelak.

Sekian untuk makluman dan tindakan tuan selanjutnya. Terima kasih.

"BERKHIDMAT UNTUK NEGARA"

Saya yang menurut perintah,

(DR. AMIR BIN MOHD SALLEH)
b.p. Pengarah,
Bahagian Perancangan dan Penyelidikan Dasar Pendidikan,
Kementerian Pendidikan Malaysia.

sk.

Pengarah Pendidikan,
Jabatan Pendidikan Negeri Perak.

Timbalan Dekan,
Sekolah Teknologi Maklumat,
UUM.

APPENDIX E

The approval letter from the Perak's State Education Department (JPN)

JABATAN PENDIDIKAN PERAK DARUL RIDZUAN,
JALAN TUN ABDUL RAZAK,
30640 IPOH,
PERAK DARUL RIDZUAN.

Telefon : 05-5274355
Fax : 05-5277273

Ruj.Kami : J.Pen.Pend.S4757/ Jld.14(24)
Tarikh : 8 Jun 2000

En. Wong Tuck Sung,
83, Lengkok Canning,
Ipoh Garden,
31400 Ipoh, Perak.

Tuan,

MEMOHON KEBENARAN UNTUK MENJALANKAN KAJIAN DAN PENYELIDIKAN DI SEKOLAH-SEKOLAH MENENGAH DALAM NEGERI PERAK.

Dengan segala hormatnya saya diarah merujuk kepada permohonan tuan yang bertarikh 8 Jun 2000 yang ada kaitan dengan Surat Kementerian Pendidikan KP(BPPDP)13/5(302) bertarikh 31 Mei 2000 mengenai perkara di atas.

2. Sukacita saya maklumkan bahawa pihak Jabatan Pendidikan Perak tiada halangan memberi kebenaran kepada tuan untuk menjalankan kajian dan pengumpulan maklumat bagi penyediaan projek yang bertajuk : **The Selection of Successful Studens for Early Admission Into Sixth Form Science Stream Using Neural Networks Model,** di sekolah-sekolah menengah negeri Perak.

Sekian dimaklumkan. Terima kasih.

'BERKHIDMAT UNTUK NEGARA'

Saya yang menurut perintah,

(HASSAN BIN IBRAHIM)
Ketua Unit Kecil,
Perhubungan & Pendaftaran,
Sektor Pengurusan Sekolah,
b.p. Pengarah Pendidikan,
Perak Darul Ridzuan.

s.k. Pengarah Pendidikan Perak.